JANUARY 1976 $1.50

BYTE

the small systems journal

Build a Light Pen

What's Inside an LSI-11?

Using Memory Address Space

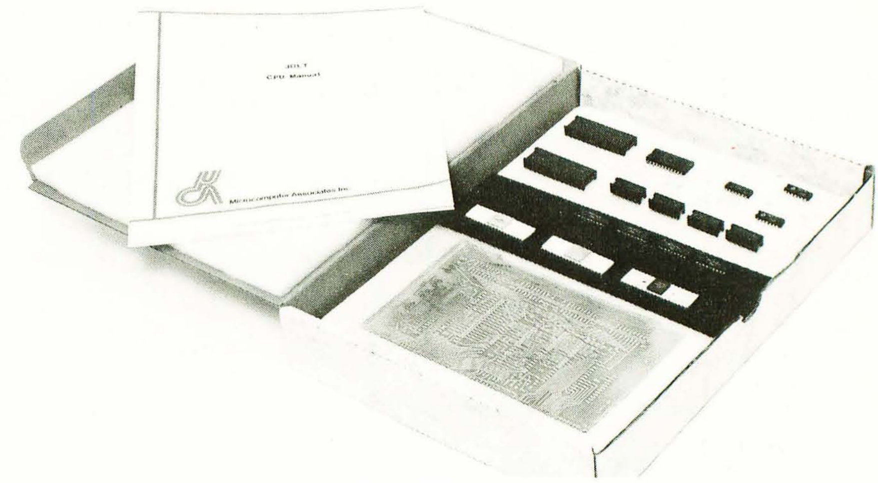
Golf Handicapping By Computer



Introducing JOLT™

...the world's lowest cost computer system

JOLT™ is the new, fully-tested microcomputer with the exclusive on-board DEMON™ debug monitor. You can build it, plug it in and talk to it in three hours or less . . . for a price of just $249!



All kits are delivered with a complete instruction manual and packaged for easy visual identification of parts to aid you in assembly.

The basic JOLT™ card includes an 8-bit MOS Technology Model 6502 CPU, which requires no clock, can directly address 65k of memory, has two index registers, 58 instructions with 11 addressing modes, two interrupts and includes both single step and address halt capability. And that's only a part of it.

JOLT's™ CPU card is available IMMEDIATELY* in either kit form or assembled ($249 for the kit in single quantity and $348 assembled). Either way, the JOLT™ CPU is completely tested prior to delivery. It comes complete with a terminal interface (TTY or EIA) and a unique software DEbugger/MONitor called DEMON™, for which full documentation is provided. It is very easy to program, and any JOLT™ delivery includes an easy-to-follow assembly instruction manual, showing you exactly how to put everything together . . . correctly. Complete assembly should take you no more than three hours if you choose the CPU in kit form. Besides the JOLT™ CPU — the 6502 from MOS Technology — the basic JOLT™ card has a fully static memory accommodating 512 bytes of the user RAM. The JOLT™ CPU memory also has 64 bytes of interrupt vector RAM. ROM Program memory on the basic card consists of 1k bytes of monitor/debugger with an automatic Power-On bootstrap program — so you can start talking to JOLT™ and it to you as soon as you plug it in to your terminal. On-board Input/Output devices on the JOLT™ CPU card include TTY 20 milliamp current loop and an EIA interface, both full duplex. The card has high speed reader interface lines and 16 fully programmable user I/O lines with full TTL drive.

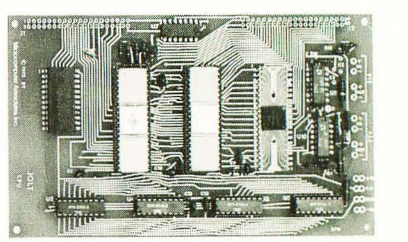
Nobody, but nobody, except MAI can offer you an on-board debugger/monitor like DEMON™. It's fully documented, too.

The exclusive DEMON™ Debug Monitor really makes JOLT™ one of the most outstanding computer systems offered at any time, at any price. Even without DEMON™ and its superior software features, JOLT™ is the lowest cost computer system in existence. And DEMON™ is a bonus you'll have to use to believe. First, it self-adapts to any terminal speed from 10-30 CPS. With it, you can display and alter your CPU register and memory locations, plus you can read, write and punch Hex formatted data . . . with Write/Punch BNPF format data for PROM programmers. It has unlimited breakpoint capability along with separate non-maskable interrupt entry and identification. External device interrupts can be directed to any location you choose, or they can be defaulted to DEMON™ recognition. DEMON™ also gives you (1) a completely protected ROM resident debug/monitor; (2) the capability to begin execution at any location in memory; (3) the capability to bypass DEMON™ entirely to permit full control by you over your system; (4) a high-speed 8-bit parallel input option; and (5) it includes user callable DEMON™ I/O subroutines. MOST IMPORTANT, DEMON™ IS INCLUDED AS STANDARD WITH ANY JOLT™ CPU KIT OR ASSEMBLED BOARD!

Obviously, the JOLT™ basic card is a computer in and of itself. But you can add significantly to its capacity and versatility by adding 4k RAM JOLT™ memories — in one card or a whole bunch. A RAM card kit is only $265 ($320 assembled). Now. *4096 Bit RAM 4K BYTE*

The JOLT™ memory card is a fully static 4,096-bit Random Access Memory (RAM) with 1 microsecond access time and on-board decoding. It is also available now.* And the quantity of one price is what you might expect to pay in quantities of 100 . . . very inexpensive!

There's also a JOLT™ I/O card for you, our Peripheral Interface Adapter. You can't beat the price — single kit 96 bucks — or the function.



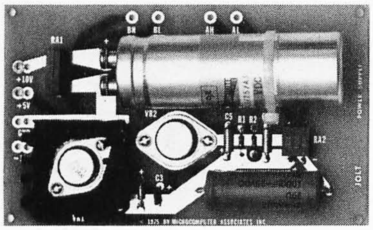
Pictured above is the assembled JOLT™ CPU card with DEMON™. just plug it in and you're ready to go.

The JOLT™ PIA (Peripheral Interface Adapter) I/O card includes two PIA LSI chips, 32 input/output lines, two interrupt lines, on-board decoding and standard TTL drive. It is also fully programmable and available IMMEDIATELY* in either kit or assembled form . . . at a very attractive single unit price ($140 assembled).

Considering the function and capacity of the JOLT™ Power Supply Card, you probably think the quantity of one price — $145 — is a misprint. It isn't.

The JOLT™ family also includes a power supply card, which operates at any of

three voltages — +5, +12 and −10. The power supply supports the basic JOLT™ CPU card, plus 4,096 bytes of RAM and I/O. The only two words for the price are "dirt cheap." It is available for delivery immediately with a single unit kit price of $145 ($190 assembled).



The assembled power supply card shown above powers the JOLT™ CPU, I/O, and RAM Memory cards.

You can also choose a blank JOLT™ universal card. Or several.

The JOLT™ Universal card is completely nude. It's a blank you can use any way you wish, for control panels, T.V. interfaces, keyboards, LED's, or any other interface logic, because the card's holes are drilled to accept 14, 16, 24, or 40 pin sockets and has the same form factor as the other JOLT™ cards. The single unit price is just $25.

If you think you need extra cables, wires and the like, choose a Super Value JOLT™ Accessory Bag. A $55 Value for just $40.

The JOLT™ Accessory Bag includes 25 separate parts, enough hardware to connect one JOLT™ card to another. Order an Accessory Bag for each additional JOLT™ card. The Bag contains such necessary items as flat cable, connectors, cord spacers, hardware, wire, etc.

THE JOLT PLAIN ENGLISH WARRANTY

All components in the JOLT™ family are new and fully tested prior to shipment. Kit components are fully warranted during the first 60 days of ownership. Assembled parts are fully warranted during the first 6 months of ownership. If your properly assembled kit does not work, just ship your order back to Microcomputer Associates Inc. and we'll repair, replace, or refund your money.

20% BONUS

NOW! Special ONE TIME BONUS DISCOUNT. Order one CPU before **November 10, 1975** and deduct 20% off of all additional cards and accessories. (Note: Discount does not apply to CPU cards, assembled or unassembled.)

We told you JOLT™ was the world's lowest priced computer system. These prices prove it.

JOLT KITS
(ALL PAYMENTS MUST BE IN U.S. DOLLARS)
QUANTITY PRICING

	1-4	5-19	20-49	50-99	100 up
JOLT™ CPU	$249	$235	$230	$225	$215
JOLT™ RAM	265	255	250	245	235
JOLT™ I/O	96	90	88	82	75
JOLT™ POWER SUPPLY	145	135	130	128	115
JOLT™ UNIVERSAL CARD	25	20	18	16	14
JOLT™ ACCESSORY BAG	40	36	34	32	30

ALL JOLT™ KITS ARE DELIVERED COMPLETE WITH A DETAILED AND ILLUSTRATED ASSEMBLY MANUAL

** Order will be shipped within 15 days.*

JOLT™ ASSEMBLED PRICES
(ALL PAYMENTS MUST BE IN U.S. DOLLARS)
QUANTITY PRICING

	1-4	5-19	20-49	50-99	100 up
JOLT™ CPU	$348	$325	$320	$315	$305
JOLT™ RAM	320	305	300	295	280
JOLT™ I/O	140	125	120	115	105
JOLT™ POWER SUPPLY	190	175	170	165	150

THE NEW JOLT™ FROM MAI IS AVAILABLE ONLY ON A DIRECT BASIS. USE THIS COUPON TODAY TO ORDER FROM

Microcomputer Associates Inc.
111 Main St., Department G
Los Altos, Calif. 94022
Phone (415) 941-1977

This coupon will bring you a JOLT™! Complete and return it.

Here is my check, money order or credit card payment for the following JOLT™ products. If a U.S. or Canadian order, please add $4.86 for shipping, handling and insurance. (Orders to all other countries are priced F.O.B. Los Altos and will be shipped FREIGHT COLLECT.)

Send me immediately:

QUANTITY	PRODUCT	PRICE
________	JOLT™ CPU KIT	$ ________
________	JOLT™ RAM MEMORY KIT	________
________	JOLT™ I/O KIT	________
________	JOLT™ POWER SUPPLY KIT	________
________	JOLT™ UNIVERSAL CARD	________
________	JOLT™ ACCESSORY BAG	________
________	JOLT™ CPU ASSEMBLED	________
________	JOLT™ RAM MEMORY ASSEMBLED	________
________	JOLT™ I/O ASSEMBLED	________
________	JOLT™ POWER SUPPLY ASSEMBLED	________
	TOTAL	$ ________
+U.S./CANADA SHIPPING/HANDLING/INSURANCE		4.86
NOTE: CALIFORNIA RESIDENTS, +6% SALES TAX		________
TOTAL REMITTANCE (U.S. DOLLARS ONLY)		**$** ________

MAIL YOUR ORDER TO: Pehaco Corporation, JOLT™ Sales Agents
Microcomputer Associates Inc. Dept. G
111 Main St., Los Altos, Ca 94022

All orders subject to the prior approval of the manufacturer.

20% BONUS NOW! Special ONE TIME BONUS DISCOUNT. Order one CPU before **November 10, 1975** and deduct 20% off of all additional cards and accessories. (Note: Discount does not apply to CPU cards, assembled or unassembled.) **20% BONUS**

PLEASE CHARGE THE TOTAL TO MY CREDIT CARD:
BANKAMERICARD # ______________________

MASTER CHARGE # ______________________
(include 4-digit #)

My credit card expires on ______________________

SIGNATURE ______________________
All credit card orders must be signed!)

I UNDERSTAND THAT ALL U.S. ORDERS WILL BE SHIPPED VIA U.P.S. AND ALL INTERNATIONAL ORDERS WILL BE SHIPPED BY AIR. SEND MY ORDER TO:

NAME ______________________

ADDRESS ______________________

ORGANIZATION ______________________

CITY ______________ STATE __________ ZIP __________

PHONE ______________________
(Include Area Code)

***Make your check payable to: Microcomputer Associates Inc.**
International Orders: Payment should be cabled to Account #00417-06355 Bank of America, 215 So. Murphy Avenue, Sunnyvale, California, USA, 94086

SOFTWARE –

■ ■ ■ ■ ■ ■ ■ is the best thing to feed your computer! Makes it healthy! Gets rid of that dull, aimless, blinking lights expression so many small systems exhibit these days. Feed your machine SOFTWARE! Make it feel good. Machine will then perform services. That will make YOU feel good. Put your machine on a good SOFTWARE diet like some of the following CHOW!

ASSEMBLER PROGRAMS FOR THE '8 0 0 8'

Discusses a "minimum length" Assembler program that can reside in 2K of memory, plus a more sophisticated version for those who have additional memory and desire a more powerful version. Included in this manual is a thorough explanation of the fundamental concepts of an assembler's operation, details on how to format the "source listing," step-by-step analysis and presentation of subroutines, program flow charts, and assembled listings of the programs! Price? A very reasonable $17.95.

AN '8 0 0 8' EDITOR PROGRAM

Describes variations of an "Editor" program that can reside in 2K of memory. Additional memory may be used to increase the size of the text buffer. The program enables one to manipulate "text" in order to create "source listings" or perform other kinds of text preparation. Includes discussion of routines, flow charts, and assembled listing. Priced at just $14.95.

'8 0 0 8' MONITOR ROUTINES

Describes a "Monitor Control" package that allows you to control the operation of your computer from an external "keyboard" device. Various routines enable you to examine and modify memory locations and CPU registers, set "breakpoints" and execute programs for "debugging" purposes, control bulk storage I/O devices, and perform other useful functions. This manual comes complete with subroutine explanations, flow charts, and an assembled, highly commented program listing. Low priced at just $11.95.

NEW! – FOR HUNGRY 8080 MACHINES!

AN '8 0 8 0' ASSEMBLER PROGRAM

This assembler program utilizes some of the unique routines we utilized in our popular '8008' assembler which enables us to provide an '8080' assembler that operates comfortably in 4K bytes of RAM (including the symbol table). An unusual feature of this assembler program is that it has been designed to accept mnemonics closely related to those used by SCELBI for our '8008' based machines. What this means is that programs originally written for an '8008' unit can be directly processed by this assembler to produce object code for an '8080' machine! NEAT! Of course, it also handles the extended instruction set of the '8080' as well. This program is provided in our popular style of a manual that discusses the major routines, presents pertinent flow charts, and includes a highly commented assembled listing. $17.95.

AN '8 0 8 0' EDITOR PROGRAM

This is essentially a "carbon copy" of the material in our earlier manual describing a '8008' Editor, except the assembled listing is provided with the machine code for an '8080.' It is a good deal at $14.95.

'8 0 8 0' MONITOR ROUTINES

These routines perform the same types of functions as described above for the '8008' version except routines were specifically developed to utilize the extended capabilities of the '8080' instruction set. Great price at just $11.95.

WANT TO "SPOON FEED" YOUR MACHINE?

You CAN learn how to develop your own machine language programs. And, if you are really serious about utilizing a small system effectively, you had better plan on learning something about it sooner or later! Here is a good way to get started.

MACHINE LANGUAGE PROGRAMMING FOR THE '8 0 0 8' (AND SIMILAR MICROCOMPUTERS)

THIS manual was written to provide the reader with the detailed knowledge one needs to know in order to successfully develop machine language programs. This information packed publication discusses and provides numerous examples of algorithms and routines that can be immediately applied to practical problems. Virtually all the techniques and routines illustrated in the manual can also be applied to other similar microcomputers such as "8080" systems (by applicable machine code conversion). The price of this exciting new manual is a low $19.95. (The floating-point arithmetic package presented in the publication is worth that price alone!)

Prices given are for domestic delivery at book mailing rate. Add $2.50 for each publication if PRIORITY air service desired (U.S.) Overseas – include $6.00 for each publication for airmail service.

(Pricing, specifications, availability subject to change without notice.)

Order direct from:

SCELBI COMPUTER CONSULTING INC.

1322 REAR BOSTON POST ROAD
DEPT DJ
MILFORD CONNECTICUT 06460

In the Queue

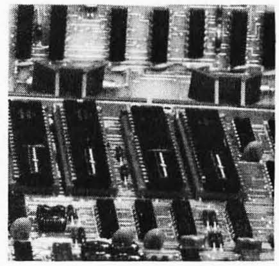
BYTE #5

JANUARY 1976

Foreground

Background

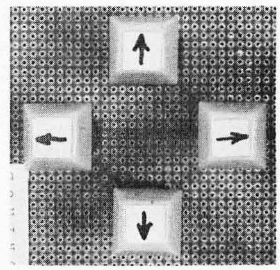
Nucleus



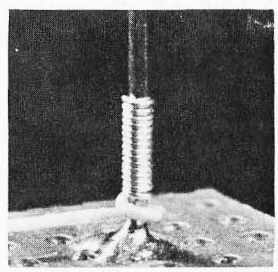
p. 12



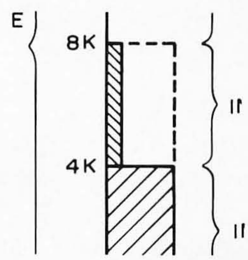
p. 26



p. 32



p. 56



p. 60

BYTE magazine is published monthly by Green Publishing, Inc., 70 Main St, Peterborough, New Hampshire 03458. Subscription rates are $12 for one year worldwide. Two years, $22. Three years, $30. Second class postage application pending at Peterborough, New Hampshire 03458 and at additional mailing offices. Phone 603-924-7217. *Address editorial correspondence to Editor, BYTE, 70 Main St, Peterborough NH 03458.*

New BYTE phone: 603-924-7217

In This BYTE

On the cover, artist Robert Tinney has provided a scene depicting the combination of computers and golf handicapping described by Dr. George Haller in his article.

How would you like a PDP-11/40 in your basement computer room? The price would probably be too high for the typical amateur. But Digital Equipment Corporation also makes the LSI-11, a microcomputer which implements the PDP-11/40 instruction set and inherits a wealth of existing PDP-11 software. Turn to Bob Baker's article on the LSI-11 for a summary.

How do you draw a picture on an oscilloscope display? **Add a Light Pen** as described by Sumner Loomis, and you will be able to add and delete points of light.

When the LIFE program can't figure out a key code, it calls DEFAULT, as described in LIFE Line 3. This issue's **LIFE Line 4** specifies the DEFAULT routine used to enter cursor motion control data and numeric data for the KEY-BOARD_INTERPRETER. As a combined hardware and software system, the LIFE application enters the realm of hardware for the first time with a simple circuit to interface the cursor motion control keyboard and an ASCII keyboard via the same input port.

Wire wrapping is a technique often used to assemble circuits. Turn to **Photographic Notes on Wire Wrapping** for some pointers for your own custom computer interfaces.

According to tradition, no computer is ever complete without blinking lights. But **There's More to Blinking Lights Than Meets the Eye,** or the control panel designer's utilitarian motives. This issue provides a few ideas for using simple and inexpensive LED indicators in ways far removed from the traditional control panel application.

In the October BYTE, Richard Gardner commented on the application of personal computers in household situations. In this issue, Ted Lau continues on that theme with an article of "structured speculation" on the **Total Kitchen Information System (TKIS).**

Computers solve problems, right? One problem which golfers have is calculating handicaps so that duffers can play against pros in the same tournament. In **Golf Handicapping (or: Buy a New Peripheral with Money Earned from Your Local Duffers),** Dr. George Haller describes a program he concocted to serve as the basis for a part time business calculating golf handicaps at his country club. For readers with teenage children, this might make a great opportunity for the kids to make some money to help pay for college expenses while learning how to run a business.

Computer systems have many resources which can be used by the person who assembles or modifies the design. One resource which is very important is the memory address space inherent in the design of the computer. **Taking Advantage of Memory Address Spaces** by James Luscher can provide important improvements in speed and function of your system.

What is style? In his article **K or k?,** Manfred Peshka describes some notational conventions which apply to BYTE's unique combination of hardware and software information. While the change of these standards is incomplete in this issue, future BYTEs will employ the standard abbreviations and units throughout.

What is one of the most useful peripherals? Why, the television set of course. To use the TV you need an interface. One such interface is the CT-1024 product by SWTPC reviewed by Jim Hogenson.

Lowest Price in the World!

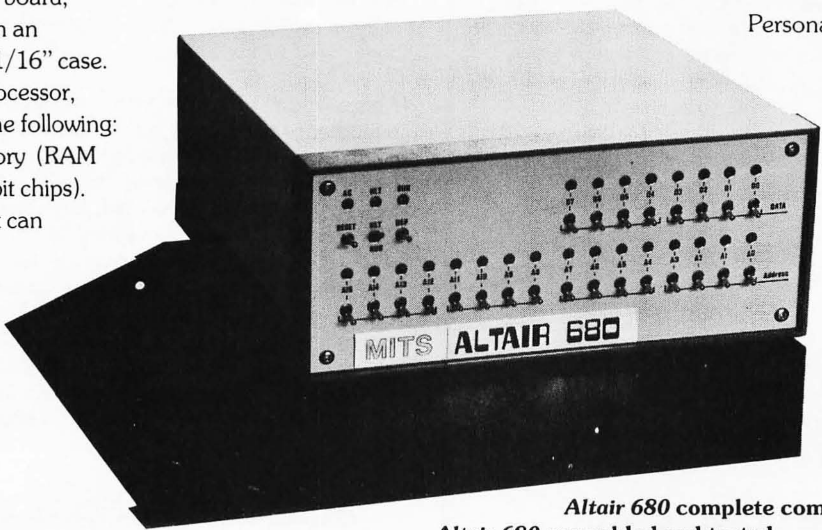
In January of 1975, MITS stunned the computer world with the announcement of the *Altair 8800* Computer that sells for $439 in kit form.

Today MITS is announcing the Altair 680.

The Altair 680, built around the revolutionary new 6800 microprocessor chip, is the lowest priced complete computer on the market. **It is now being offered at the special, low price of $345!**

The Altair 680 comes with power supply, front panel control board, and CPU board inclosed in an 11" wide x 11" deep x 4 11/16" case. In addition to the 6800 processor, the CPU board contains the following:

1. 1024 words of memory (RAM 2102 type 1024 x 1-bit chips).
2. Built-in Interface that can be configured for RS232 or 20 mA Teletype loop or 60 mA Teletype.
3. Provisions for 1024 words of ROM or PROM.



The Altair 680 can be programmed from the front panel switches or it can be connected to a computer terminal (RS232) or a Teletype such as an ASR-33 or surplus five-level Baudott Teletype (under $100).

The Altair 680 can be utilized for many home, commercial or industrial applications or it can be used as a development system for *Altair 680* CPU boards. With a cycle time of 4 microseconds, 16-bit addressing, and the capability of directly addressing 65,000 words of memory and a virtually unlimited number of I/O devices, the *Altair 680* is a very versatile computer!

Altair 680 Software

Software for the *Altair 680* includes a monitor on PROM, assembler, debug, and editor. This software will be available to *Altair 680* owners at a nominal cost.

Future software development will be influenced by customer demand. MITS will sponsor lucrative software contests to encourage the rapid growth of the *Altair 680* software library. Programs in this library will be made available to all *Altair 680* owners at the cost of printing and mailing.

Altair Users Group

All *Altair 680* purchasers will receive a free one year membership to the Altair Users Group. This group is the largest of its kind in the world and includes thousands of *Altair 8800* and *680* users.

Members of the Altair Users Group are kept abreast of Altair developments through the monthly publication, ***Computer Notes.***

Altair 680 Documentation

The *Altair 680* kit comes with complete documentation including assembly manual, assembly hints manual, operation manual, and theory manual. Assembled units come with operation and theory manuals. Turnkey model and CPU boards also include documentation.

NOTE: *A complete set of Altair 680 manuals in a 3-ring Altair binder is now available for $14.50 (regularly $25). Offer expires January 30, 1976.*

Delivery

Personal checks take 2-3 weeks to process while money orders and credit card purchases can be processed in 1-3 days. Delivery should be 30-60 days but this can vary according to order backlog. All orders are handled on a first come, first served basis.

Altair 680 Prices

Altair 680 **complete computer kit**	**$345**
Altair 680 **assembled and tested**	**$420**
Altair 680T **turnkey model (complete** ***Altair 680*** **except front panel control board) Kit Only**	**$280**
Altair 680 **CPU board (including pc board, 6800 microprocessor chip, 1024 word memory, 3 way interface and all remaining components except power supply)**	**$195**
Altair 680 **CPU board assembled and tested**	**$275**
Option IC socket kit (contains 40 IC sockets. CPU, memory and PROM sockets come with 680 kit)	**$ 29**
Option cooling fan (required when expanding 680 internally)	**$ 22**
Option cooling fan installed	**$ 26**
PROM kit (256 x 8-bit ultraviolet, erasable 1702 devices)	**$ 25**
Connectors (Two sets of 25-pin connectors. Required when interfacing 680 to external devices)	**$ 22**

Prices, delivery and specifications subject to change.

MITS

"Creative Electronics"

MITS/6328 Linn N.E., Albuquerque, NM 87108 505/262-1951

MAIL THIS COUPON TODAY!

☐ Enclosed is check for $__________
☐ BankAmericard #__________ ☐ or Master Charge #__________
☐ Altair 680 ☐ Kit ☐ Assembled ☐ Altair 680T ☐ Altair 680CPU
☐ Socket Kit ☐ Fan ☐ PROM kit ☐ Documentation Special
☐ Please send literature

NAME____________________

ADDRESS____________________

CITY________________ STATE & ZIP__________

MITS/6328 Linn N.E., Albuquerque, NM 87108 505/262-1951

CALL FOR PAPERS-ACM '76 OCTOBER 20-22,1976...

HOUSTON

Our 1976 Annual Conference will be held October 20 — 22 in Houston. We feel that it will be one of the most relevant ACM conferences ever held, both for the practicing professional and the researcher. The conference facilities in downtown Houston's Regency Hyatt House will be outstanding. You will find that ACM '76 will be a place where you can enjoy yourself, renew old acquaintances, make some new ones, and participate in a serious technical discussion of the problems and potential of our industry. Plan to attend.

Olin Johnson
General Chairman, ACM '76

about the conference

You are invited to submit a paper on any aspect of computer research, development, or implementation. High quality papers of a theoretical, state-of-the-art or tutorial nature are welcome. The technical program for ACM '76 will again be organized around the ACM Special Interest Groups, although additional sessions will be provided for papers of general interest or those not related to any SIG.

Papers must not have been previously presented or published; they should not exceed 10 published pages, including a 100-word abstract, bibliography, and illustrations; and they must be received in four copies by March 1, 1976. All papers will be refereed either by the SIG's or by reviewers selected by the Technical Program Committee. Notification of acceptance will be made by May 1. If you intend to submit a paper, and we sincerely hope that you do, please send in the coupon below. If you have an idea for a good session or some other technical program idea, please do the same or start promoting your idea in your SIG —— but do it now! Be a part of the spirit of ACM '76!

about the technical program

ACM

Be a part of the "Spirit of ACM '76"

ASSOCIATION for COMPUTING MACHINERY
1976 ANNUAL CONFERENCE

Roger W. Elliott
Technical Program Chairman, ACM '76
Department of Industrial Engineering
Texas A & M University
College Station, Texas 77843 (713) 845-5531

Name ______________________ Phone __________

Address ______________________________

City ______________ State __________ Zip ______

☐ I intend to submit a technical paper.

Tentative title ______________________________

______________ SIG (optional) ______________

☐ I am enclosing a technical program suggestion.

☐ Please call me. I would like to discuss an idea for the technical program.

MEMORIES ARE MADE OF THESE:

INTEGRATED CIRCUITS

2102 1K STATIC RAM---STILL ONLY $1.95 GUARANTEED FASTER THAN 750 NS. BUY 100, TAKE 20%; BUY 1000, TAKE 30%.

5203 2K EROM --- $9.50. THIS STATIC MEMORY IS FULLY PROGRAMMABLE AND MAY BE ERASED WITH UV LIGHT. NON-VOLATILE.

5204 4K EROM---$24.50. SIMILAR TO THE 5203, BUT 4K OF MEMORY.

MEMORY/PROCESSOR 8 BIT CHIP SET: $32.50 BRINGS YOU 8 OF OUR 2102s AND AN 8008 MICROPROCESSOR.

EROM BOARD KITS

8K BY 8 BOARD...$352 (ADD $48 & WE PROGRAM)
4K BY 8 BOARD...$200 (ADD $25 & WE PROGRAM)
2K BY 8 BOARD...$125 (ADD $15 & WE PROGRAM)

Now you can stuff up to 8K of your favorite software----editor/assembler or what have you----on a board. Program it yourself or have us do the programming. Just like the RAM kit, DIRECT ALTAIR 8800 PLUG-IN COMPATIBLE. Includes sockets, bypassing, industrial - quality plated - through board, buffered addresses and outputs, on board regulation. LOW POWER: Full 8K requires ½A @ 5V, 150 ma @ -12V. File protect feature includes line receiver with real hysteresis to prevent false triggering on noise. EXPANDABLE: If you buy our 2K board and want to move on to bigger and better things (like 8K), all you add is more sockets and EROMS. Your board doesn't become obsolete.

4K BY 8 RAM BOARD $109.22

ONE-THIRD OF A CENT PER BIT --- Our most popular memory kit. You get sockets for all ICs, industrial-quality plated-through board, lots of bypassing, 5 voltage regulators to share the power load (less thermal problems, more reliability), typical 500 ns access time at 25°C, and buffered addresses and outputs (no input presents more than 1 LPTTL load; outputs can drive 20 standard TTL loads). May be powered with 8 volts unregulated; or by bypassing the on board regulators, with +5V. Complete with assembly instructions and logic print of the RAM board.

ALTAIR 8800 OWNERS, PLEASE NOTE: THIS MEMORY BOARD KIT, LIKE OUR OTHER MEMORY KITS, IS DIRECTLY PLUG-IN COMPATIBLE WITH THE 8800.

EROM BOARD SPECIAL OFFER

We've got 4K by 8 boards pre-programmed with 8080 assembler, editor, and monitor routines. $200 buys you the same quality and ALTAIR PLUG-IN COMPATIBILITY as our other memory kits. Another advantage: because this is EROM rather than mask programming, you can make changes if desired. Speaking of changes, although these routines work our really well, we are including 2 free updates to purchasers should bugs or modifications develop. Any additional changes are available for a very nominal charge. SEND $2.95 (PPD) FOR OUR SOFTWARE INFO PACKET IF YOU WANT MORE INFO OR WANT TO LOAD YOUR OWN ROM BOARD. INCLUDES INSTRUCTION LISTING, SCHEMATIC, ASSEMBLY DATA, ETC.

LOW POWER SCHOTTKY

CAVE grafix

COMPARED TO STANDARD TTL, LOW POWER SCHOTTKY OFFERS 1/5 THE POWER CONSUMPTION-BUT WITH NO SPEED PENALTY. CONSIDERED BY MANY AS THE LOGICAL SUCCESSOR TO STANDARD TTL.

74LS00	Quad 2 input NAND gate	.40
74LS04	Hex inverter	.45
74LS08	Quad 2 input AND gate	.40
74LS10	Triple 3 in NAND gate	.40
74LS20	Dual 4 input NAND gate	.40
74LS42	BCD to decimal decode	1.25
74LS138	1 of 8 decoder	1.40
74LS168	Dec Syn Up/Down cntr	1.50
74LS169	Bin Syn Up/Down cntr	1.50
74LS175	Quad latch	1.38

Hard to find TTL

74273	8 bit latch (20 pin)	1.89
74365	Noninv 3 st buf NOR en	.63
74366	Inv 3 state buf NOR en	.63
74367	Noninv 4-2 buf w/ sep en	.63
74368	Inv 4-2 buf w/ sep en	.63
74390	16 pin dual decade cnt	1.50
74393	14 pin dual binary cntr	1.50

MICROPROCESSORS

PACE

16 BIT CHIP SET $195

1 • PACE CPU
4 • DS3608
1 • DS0026
1 • DM8837
32 • 2102 (4k bytes RAM)
2 • MM5204 (1k byte ROM)
AND OUR PACE DATA PACKET!

A POWERFUL 16 BIT PROCESSOR. 40 PIN DIP; CAPABILITY FOR 45 CLASSES OF INSTRUCTIONS AND UP TO 337 INSTRUCTIONS. INTERFACE CHIPS PROVIDE TTL COMPATIBILITY--RAMS AND ROMS PROVIDE MEMORY, AND THE PACE DATA PACKET TELLS YOU HOW TO APPLY EVERYTHING.

PACE DATA PACKET AVAILABLE SEPARATELY FOR $2.50 (COVERS PRINTING AND POSTAGE). 80 PAGES OF INFORMATION ON THE CHIP ITSELF, SOFTWARE, SYSTEM ORGANIZATION, AND MORE. ALSO INCLUDES AN 11" BY 17" INCH FOLDOUT LOGIC PRINT OF PACE IMPLEMENTATION.

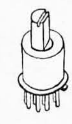
IF YOUR INTEREST IS MICROCOMPUTERS, AND PARTICULARLY PACE, YOU WILL WELCOME THIS DATA PACKET. PURCHASE PRICE REFUNDABLE WITH PACE ORDER.

GODBOUT

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

TERMS: Add postage where indicated; add 50¢ to orders under $10. No COD orders; it's a lot of paperwork for us. Californians add tax. To place Bankamericard® or Mastercharge® orders, call (415) 357-7007, 24 hours a day. Please remember this is an answering service, and is not set up to handle technical questions.
Want to know about all the other stuff we sell? Send for our FREE FLYER!

TINY Rotary Switch



SINGLE POLE, SEVEN POSITION + OFF. IN TO-5 CAN; TINY FOR CIRCUIT BOARD MOUNTING. GOOD FOR PROGRAMMING & MUCH MORE. $1.25 EA OR 3/$2.95

SON OF A CHEAP CLOCK $14.50

THIS IS THE SUCCESSOR TO OUR 'CHEAP CLOCK' KIT. IT IS AN EASY TO ASSEMBLE KIT THAT INCLUDES EVERYTHING YOU NEED TO MAKE A WORKING CLOCK, EXCEPT CASE AND HARDWARE.

IT FEATURES 3/10 IN. READOUTS FOR HOURS, MINUTES, AND SECONDS. 50/60 HZ OPERATION, 12 OR 24 HOUR DISPLAY FORMAT.

SOCKETS FOR IC AND READOUTS PROVIDED.

THE READOUTS CAN BE EASILY REMOTED; THE CLOCK, IN CONJUNCTION WITH OUR 60 HZ TIME BASE ($10.95) IS AN EXCELLENT AUTOMOBILE CLOCK.

ADD SHIPPING FOR 1 POUND.

BOTTOM-OF-THE-PAGE-ITEMS:
LM340T: 1A regulator, TO-220 plastic package. 6 or 8 volts: 95¢ ea.

Beach Ball Software

Editorial
by
Carl Helmers

What a predicament. You've bought this box full of printed circuit boards, transformer iron and integrated circuit silicon. After long hours paying attention to the details of an intricate assembly process it is "done." To the best of your abilities you've verified that the box does what it is supposed to do. Now it's sitting over there on the bench (or living room table (or office desk)) plugged into the AC wall outlet and grinning like a Cheshire cat with a mouth full of LED or TVT teeth. So, now what do you do with it? Like many objects of some bulk and substance, it seems so useless – but is it? There are lots of human designed objects which at first sight appear to be useless. But most turn out to have great potential value which can be released by the active and creative human mind. The secret of use and enjoyment is the application of that most characteristically human capacity to think. The hobby created about these artificial "brains" is an excellent training ground for the human brain – one of the secrets of its appeal.

But Then, What Good is a Beach Ball Either?

As an example of a simple object which requires creativity to realize its potential, consider a child's beach ball. Imagine for a moment that your newly constructed computer kit is sitting on its pedestal and begins slowly to change into a beach ball through some advanced technological "magic." Its corners begin to round, the edges smooth out, and after some moments of transformation it is a jolly round beach ball having the same general properties the computer had: it sits there looking useless to the skeptic.

As an intelligent and active being, curiosity is one of your most admirable traits. So you walk over to the pedestal, pick up that beach ball and begin a thorough examination of its characteristics. What do you find? As you pick it up you verify that it is a solid object which has the expected weight of a beach ball. You rotate it about several axes and verify that it is indeed a sphere within the limits of your perception. Using an appropriate measuring tool you find that its diameter is 30 cm. The surface is resilient but hard. The ball slips out of your hands and drops to the floor where its resilience is confirmed by a healthy rebound. After the examination, you put it back on the pedestal. Again, the thought: what good is a beach ball? It sits there with no life of its own, not even condescending to wink an LED at you. How could anyone possibly shell out hard earned cash for such an object?

Beach Ball Software

But people do shell out lots of hard earned money for beach balls. And people such as you and I are not fools when we buy a quality beach ball. We know beach balls are useful through our appreciation of fun and advertising of the products. The beach ball is the basic "hardware" required to play numerous games; beach balls are even used on occasion for more practical tasks. The beach ball is a widely distributed standard product with a simple design. Very little user documentation is supplied by the manufacturers and distributors of beach balls. The wide acceptance of the beach ball is due to the numerous applications inherent in its design. These applications await the creative minds of human beings who decide what to do with the ball and how to do it. Any game or recreation which uses the beach ball as a prop is an example of the "beach ball software." Such games may have been made up generations ago and stored in the form of word of mouth learning, printed books or magazines on the subject. Or, they may be made up or reinvented spontaneously and never recorded at all. Whatever the source, the game and its rules form a method of using the beach ball which takes advantage of its properties, but is not in any way built into the device.

So having temporarily approached the problem from the point of view of a beach ball, let's invert the transformation and turn the object back into a computer. We still

A manufacturer who supplies a series of interesting sample programs to illustrate the use of his processor and system is providing you with a valuable service . . .

have not solved the problem of where to get the specific creativity required for the computer software, but the beach ball analogy provides a useful model.

A computer system, like a beach ball, is an incomplete system no matter how fully supported by manufacturers' software. No manufacturer can tell you what you want to do with the machine in complete detail. The manufacturer can supply suggested uses and build in features which make certain uses inherently easy to program, but ultimately the user of the computer must decide what to program or when to use the programs available. The beach ball system is completed when the beach ball user decides upon a particular game or practical application in which the beach ball is a key element. It might be that the beach ball simply makes the application easier to perform, or it might be that the beach ball is so crucial that the application would not even have been possible without that ball. The computer system is also completed when you – the user – decide how you want to employ the computer. It might be that you use the computer simply to improve the efficiency of some menial task, or you might use the computer in an application – such as complex interactive games – which would have been impossible to perform minus the computer's intelligence. Like any activity, you get out of a computer rewards which are in proportion to the effort applied to understand and "get into" the activity.

With computers – as with beach balls – it is often possible to find a starting point in this area of applications and software. Creativity of the individuals who have previously thought about the subjects involved can often be useful as a starting point. By seeing concrete examples of the uses of the computer systems, you can acquire a valuable basic understanding of what can be done. One obvious source of such information is in the form of BYTE articles.

Another form is in the user support packages provided by manufacturers. A manufacturer who supplies a series of interesting sample programs to illustrate use of his processor and system is providing you with a valuable service – you find out how things are done with his computer in specific and detailed examples. One excellent example of such a service is the documentation which can be purchased for the assembler and text editor programs of the SCELBI computers – complete listings of the programs with detailed explanations and design information are provided in the manuals for these SCELBI program products. When evaluating a computer system, look for such examples in the documentation and compare the offerings on this point. There are other sources of information about the creative uses of the computers of the world. Direct information can often be acquired in the context of the computer club, an association of users for the purpose of solving problems, exchanging ideas, filling in gaps in knowledge, etc. When you talk to your technological neighbor about what he or she is doing, you can get valuable information on the possible applications of your machine; and of course your neighbor receives similar benefits in return.

It is this kind of exchange of creativity and ideas among interested individuals which will make the personal computing field into a well developed facet of our civilization, growing out of the scattered and isolated efforts of individuals who pioneered the activity. One of the main reasons for BYTE magazine's very existence is to help foster this transfer of information, to give our readers the knowledge and ideas useful as creative starting points for personal applications of computers. ■

BOMB: BYTE's Ongoing Monitor Box

BYTE would like to know how readers evaluate the efforts of the authors whose blood, sweat, twisted typewriter keys, smoking ICs and esoteric software abstractions are reflected in these pages. BYTE will pay a $50 bonus to the author who receives the most points in this survey each month. (Editor Helmers is not eligible for the bonus.) The following rules apply:

1. *Articles you like most get 10 points, articles you like least get 0 (or negative) points – with intermediate values according to your personal scale of preferences.*
2. *Use the numbers 0 to 10 for your ratings, integers only.*
3. *Be honest. Can all the articles really be 0 or 10? Try to give a preference scale with different values for each author.*
4. *No ballot box stuffing: Only one entry per reader!*

Fill out your ratings, and return it as promptly as possible along with your reader service requests and survey answers. Do you like an author's approach to writing in BYTE? Let him know by giving him a crack at the bonus through your vote.

Page No.	Article	LIKED LEAST … BEST
12	Baker: Mini-Microcomputer	0 1 2 3 4 5 6 7 8 9 10
26	Loomis: Light Pens	0 1 2 3 4 5 6 7 8 9 10
31	Warren: Horror Story	0 1 2 3 4 5 6 7 8 9 10
32	Helmers: LIFE Line 4	0 1 2 3 4 5 6 7 8 9 10
42	Lau: Kitchen Information System	0 1 2 3 4 5 6 7 8 9 10
46	Haller: Golf Handicapping	0 1 2 3 4 5 6 7 8 9 10
50	Dittrich: Op Code Table	0 1 2 3 4 5 6 7 8 9 10
52	Helmers: Blinking Lights	0 1 2 3 4 5 6 7 8 9 10
56	Helmers: Wire Wrapping	0 1 2 3 4 5 6 7 8 9 10
60	Luscher: Memory Address Space	0 1 2 3 4 5 6 7 8 9 10
64	Peshka: K or k?	0 1 2 3 4 5 6 7 8 9 10
92	Hogenson: CT-1024 Kit Review	0 1 2 3 4 5 6 7 8 9 10

A New Mini-Microcomputer System

The Digital Equipment Corporation LSI-11

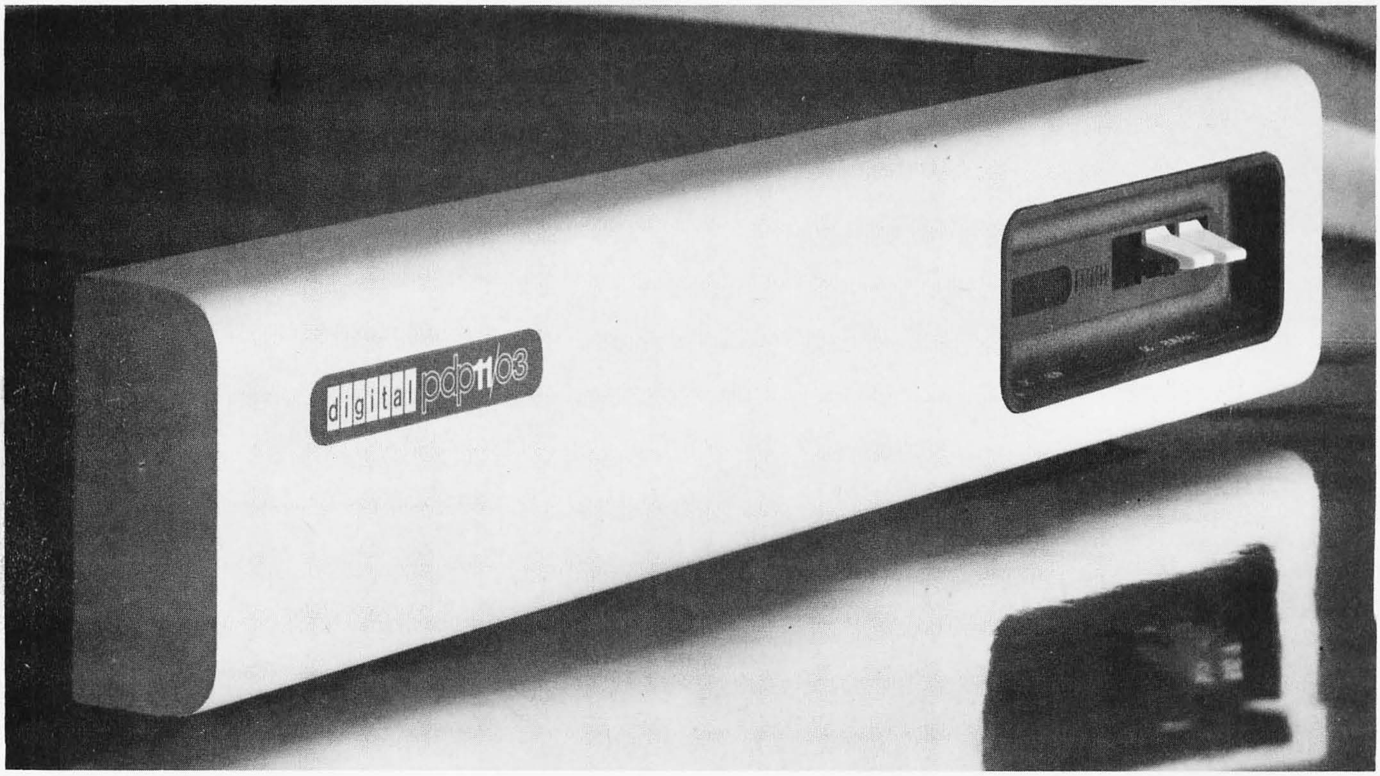
Robert W. Baker
34 White Pine Dr.
Littleton MA 01460

Digital Equipment Corporation has a new addition to the microcomputer market. Designated the LSI-11, it is a complete 16 bit microcomputer system on a single 8.5 inch by 10 inch (21.6 cm by 25.4 cm) printed circuit board, combining the instruction set of a PDP-11/40 with an under $1000 price.

A 3.5 inch H by 19 inch W by 13.5 inch D (8.9 cm by 48.3 cm by 34.3 cm) boxed version of the LSI-11 is designed as an off-the-shelf microcomputer system. Designated the PDP-11/03, it consists of an LSI-11 microcomputer, serial line interface, power supply, and a mounting box designed to mount in a standard 19 inch cabinet. Removing the front panel exposes the LSI modules and cables allowing replacement or installation of a module from the front of the PDP-11/03. The power supply has three front panel switches and indicators accessible through a cutout in the front panel. The lights and switches are still attached to the power supply and functional when the front panel is removed. Input power of the PDP-11/03 is typically 190 Watts at full load.

LSI-11 Evolution

The processor, memory, device interfaces, backplane and interconnecting hardware of the LSI-11 are all modular in design to allow custom tailoring necessary for specific application requirements. It was not intended to be a low end minicomputer, but to provide minicomputer capability to the new microcomputer applications.



To accomplish this goal, the LSI-11 was designed to optimize system costs rather than component costs. A four-chip microprogrammed central processor was selected to emulate the PDP-11 instruction set, allowing the inclusion of automatic dynamic memory refresh without additional cost. The microprogrammed processor also makes feasible user microcode and an ASCII console which will be discussed later.

Central Processor

The central processor module consists of the microprogrammed processor and 4096 words of memory, together with the bus transceivers and control logic. The four chip microcomputer controls the time allocation of the LSI-11 bus for peripherals and performs all arithmetic and logic operations as well as instruction decoding. Eight 16 bit, general-pupose registers can be used as accumulators, address pointers, index registers, stack pointers, or other desired functions. Arithmetic operations can be from one register to another, from one memory location or device register to another, or between a memory location or a device register and a general register. Data transfers between IO devices and memory on the bus occur without disturbing the processor registers.

Bus

The bus, which is implemented on the H9270 card guide backplane assembly, is the data path which enables a complete system to be configured. This bus was designed to allow low cost peripheral interfaces for microcomputer applications, rather than to support the wide range of peripheral configurations common to large minicomputer systems. The processor module is capable of driving six device slots along the bus without additional termination, as provided with the H9720 backplane. Devices or memory can be installed in any location along the bus, as most bus control and data signals are bidirectional, open-collector lines that are asserted when low. The bus signals include 16 multiplexed data/address lines, 6 data transfer control lines, 6 system control lines, and 5 interrupt and direct memory access (DMA) control lines.

Any communication between two devices on the bus is in the form of a master-slave relationship. Only one device, the bus master, can have control of the bus at any point in time. The master device controls the bus while communicating with another device on the bus, the slave. Since the LSI-11 bus is used by the processor and all IO devices, there is a priority structure to determine which device gets control of the bus. Every device on the bus capable of becoming bus master has a specific priority associated with its position along the bus. When two devices request use of the bus simultaneously, the higher priority device will receive control. All data transfers on the bus are interlocked so that communication is independent of the physical length of the bus and the response time of the slave so long as a bus timeout does not occur. Asynchronous operation allows each device to operate at the maximum possible speed.

Interrupt System

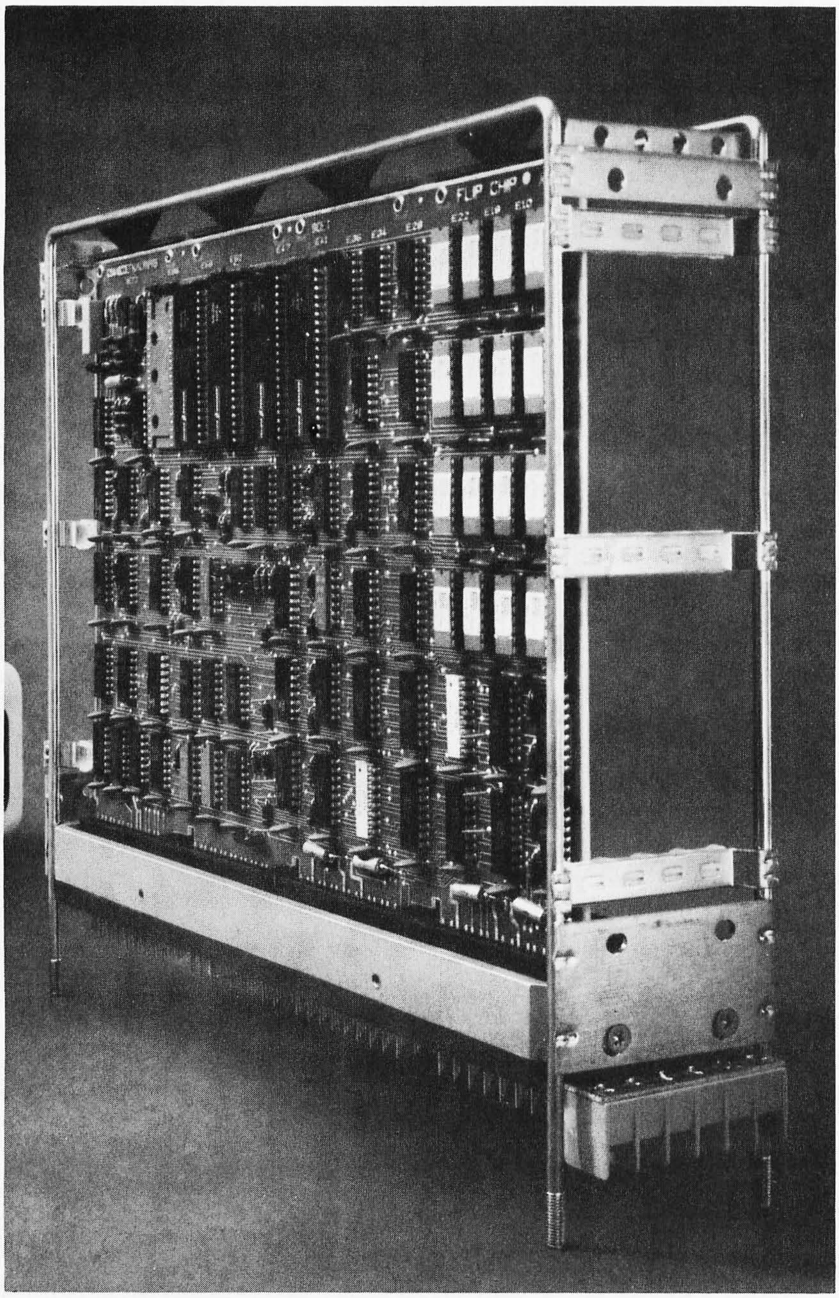
Interrupt and DMA handling incorporates two daisy-chained grant signals. This method eliminates device polling to service interrupt requests and establishes an interrupt priority. The highest priority device is the module located electrically closest to the microcomputer module. Only when a device is not asserting a request does it pass grant signals to lower priority devices. When an interrupting device receives a grant, the device passes to the processor an interrupt vector which points to a new processor status word (PSW) and the starting address of an interrupt service routine for the device. The current value of the PSW and program counter (PC) are stored on the stack.

The processor operates with the interrupt mask (PSW bit 7) set (1) or cleared (0). When PSW bit 7 is equal to 1, no external device can interrupt the processor with a request for service. The processor must be operating at PSW bit 7 equal to 0 for the device's request to be effective. Interrupts can occur only between processor instructions since they change the state of the processor. DMA operations, on the other hand, may occur between individual bus cycles since these operations do not change the processor state.

One signal line on the bus functions as an external event interrupt line to the processor module. When connected to a 60 Hertz line frequency source, this signal line can be used as a real-time clock interrupt. When automatic interrupt dispatch (vectoring) is not needed, this line may be used as a common interrupt signal. Although this necessitates device polling (as in earlier computers, such as the PDP-8), device interfaces may now be slightly less complicated. A single connection on the processor module enables or inhibits the external event interrupt. When enabled, the device connected to this line has a higher interrupt priority than any device connected to the daisy-chained grant signals.

Power Fail/Restart

To further increase the system flexibility, several power fail/restart options are avail-



able. The power fail sequence is initiated upon sensing a warning signal from the power supply signaling an impending AC power loss. The current PSW and PC are pushed on the processor stack and a new PC and PSW are taken from a vector at location 24. Normally, with non-volatile memory, this routine would save processor registers, set up a restart routine, and halt. When only volatile memory is used, the registers cannot be saved but the power fail trap does allow an orderly system shutdown to occur.

When AC power is restored, one of the four jumper selectable power-up options is initiated. The first option is loading a programmed PSW and PC from the vector at location 24. This would be used with non-volatile memory to continue execution of the program at the point where the power fail occurred or to restart the program at an arbitrary address with ROM program storage. If the BHALT line on the bus (the halt switch) is asserted during this power-up sequence, the ASCII console microcode will be entered immediately after loading the PSW and PC. The second power-up option causes an unconditional entry to the ASCII console routines. The processor can then be started by an ASCII console command allowing remote system starting without controlling the bus halt line. (More on the ASCII console later.) Alternately, the last two options allow program execution to begin at a specified address in either macrocode or microcode.

Memory

The 4096 word memory on the basic CPU module consists of sixteen 4096 bit dynamic RAMs. This memory logically appears on the external bus while being physically on the CPU module. Being accessible to the bus allows external DMA transfers to take place to and from the basic 4096 word memory. Also, an optional jumper allows the CPU module memory to occupy either the first or second 4096 word block of the bus address space.

Various memory modules are available for applications requiring more storage than the standard 4096 word MOS memory on the processor board. Those offered include a non-volatile 4096 word core memory, a 1024 word static RAM, read-only memory (PROM/ROM) with a maximum capacity of 4096 words per board in 512 word increments or 2048 words in 256 word increments, and a 4096 word dynamic MOS RAM.

A common disadvantage of using dynamic MOS memory is the necessity of refreshing the contents of memory at specific intervals. The refresh operation is required to replace the stored charge in each

The MODULAR MICROS from MARTIN RESEARCH

Here's why the new *MIKE 2* and *MIKE 3* are the best values in microcomputers today!

8008 OR 8080

Martin Research has solved the problem bothering many potential micro users . . . . whether to go with the economical 8008 microprocessor, or step up to the powerful 8080. Our carefully designed bus structure allows either processor to be used in the same system!

The *MIKE 3* comes with an 8080 CPU board, complete with crystal-controlled system timing. The *MIKE 2* is based on the 8008. To upgrade from an 8008 to an 8080, the user unplugs the 8008 CPU board and plugs in the 8080 CPU. Then he unplugs the 8008 MONITOR PROM, and plugs in the 8080 MONITOR PROM, so that the system recognizes the 8080 instruction set. That's about it!

If the user has invested in slow memory chips, compatible with the 8008 but too slow for the 8080 running at full speed, he will have to make the 8080 wait for memory access—an optional feature on our boards. Better still, a 4K RAM board can be purchased from Martin Research with fast RAM chips, capable of 8080 speeds, at a cost no more that you might expect to pay for much slower devices.

In short, the *MIKE 2* user can feel confident in developing his 8008 system with expanded memory and other features, knowing that his *MIKE 2* can be upgraded to a *MIKE 3*—an 8080 system—in the future.

EASE OF PROGRAMMING

Instructions and data are entered simply by punching the 20-pad keyboard. Information, in convenient octal format, appears automatically on the seven-segment display. This is a pleasant contrast to the cumbersome microcomputers which require the user to handle all information bit-by-bit, with a confusing array of twenty-odd toggle switches and over thirty red lights!

A powerful MONITOR program is included with each microcomputer, stored permanently in PROM memory. The MONITOR continuously scans the keyboard, programming the computer as keys are depressed.

Say the user wishes to enter the number *135* (octal for an 8008 OUTPUT 16 instruction). He types *1,* and the right-hand three digits read *001.* Then he presses *3,* and the digits say *013.* Finally he punches the *5,* and the display reads *135.* Notice how the MONITOR program

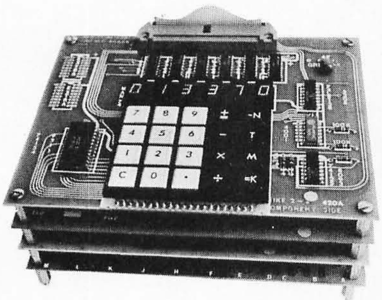
(Continued in column 3.)

QUICK....
what number is this?

●○ ○●○ ●○●

If you have to read your microcomputer like this--bit by bit, from rows of lights--the computer's making *you* do *its* work. And if you have to use rows of toggle switches to program it, you might wonder why they call the computer a labor-saving device!

Contrast the layout of a typical pocket calculator. A key for each number and function; six easy-to-read digits. Why not design microcomputers like that?



Here they are! The *modular micros* from Martin Research. The keyboard programs the computer, and the bright, fully-decoded digits display data and memory addresses. A Monitor program in a PROM makes program entry easy. And, even the smallest system comes with enough RAM memory to get started!

Both the *MIKE 2* system, with the popular 8008 processor, and the 8080-based *MIKE 3* rely on the *same* universal bus structure. This means that accessories--like our 450 ns 4K RAM--are compatible with these and other 8-bit CPUs. And, systems start at under $300! For details, write for your...

FREE CATALOG!



MIKE 2 MANUAL... This looseleaf book includes full information on the *MIKE 2* system, with schematics.

Price for orders received by November 15, 1975... **$19**

Includes a certificate worth $10 towards a *modular micro* system, good 90 days. *(Offer valid, USA only.)* After 11/15: $25.

modular micros | **martin research**

Martin Research / 3336 Commercial Ave.
Northbrook, IL 60062 / (312) 498-5060

shifts each digit left automatically as a new digit is entered! The value on the display is also entered into an internal CPU register, ready for the next operation. Simply by pressing the *write* key, for example, the user loads *135* into memory.

The MONITOR program also allows the user to step through memory, one location at a time (starting anywhere), to check his programming. Plus, the Swap Register Option allows use of the interrupt capabilities of the microprocessor: the MONITOR saves internal register status upon receipt of an interrupt request; when the interrupt routine ends, the main program continues right where it left off.

We invite the reader to compare the programmability of the *MIKE* family of microcomputers to others on the market. Notice that some are sold, as basic units, *without any memory capacity at all.* This means they simply cannot be programmed, until you purchase a memory board as an "accessory." Even then, adding RAM falls far short of a convenient, permanent MONITOR program stored in PROM. Instead, you have to enter your frequently-used subroutines by hand, each and every time you turn the power on.

EASY I/O INTERFACE

The *MIKE* family bus structure has been designed to permit easy addition of input and output ports. A hardware interface to the system generally needs only two chips—one strobe decoder, and one latching device (for output ports) or three-state driving device (for inputs). A new I/O board can be plugged in anywhere on the bus; in fact, all the boards in the micro could be swapped around in any position, without affecting operation. I/O addresses are easy to modify by reconnecting the leads to the strobe decoder (full instructions are provided); this is in marked contrast to the clumsy input multiplexer approach sometimes used.

POWER & HOUSING

The micros described to the left are complete except for a cabinet of your own design, and a power supply. The basic micros require +5 V, 1.4 A, and −9 V, 100 MA. The 4K RAM board requires 5 V, 1 A. A supply providing these voltages, and ±12 V also, will be ready soon.

OPTIONS

A number of useful micro accessories are scheduled for announcement. In addition, the *MIKE 3* and *MIKE 2* may be purchased in configurations ranging from unpopulated cards to complete systems. For details, phone, write, or check the reader service card.

memory cell which has been lost through leakage currents. To eliminate most of the control circuitry normally necessary to perform this memory refresh, the LSI-11 CPU microcode features automatic refresh control.

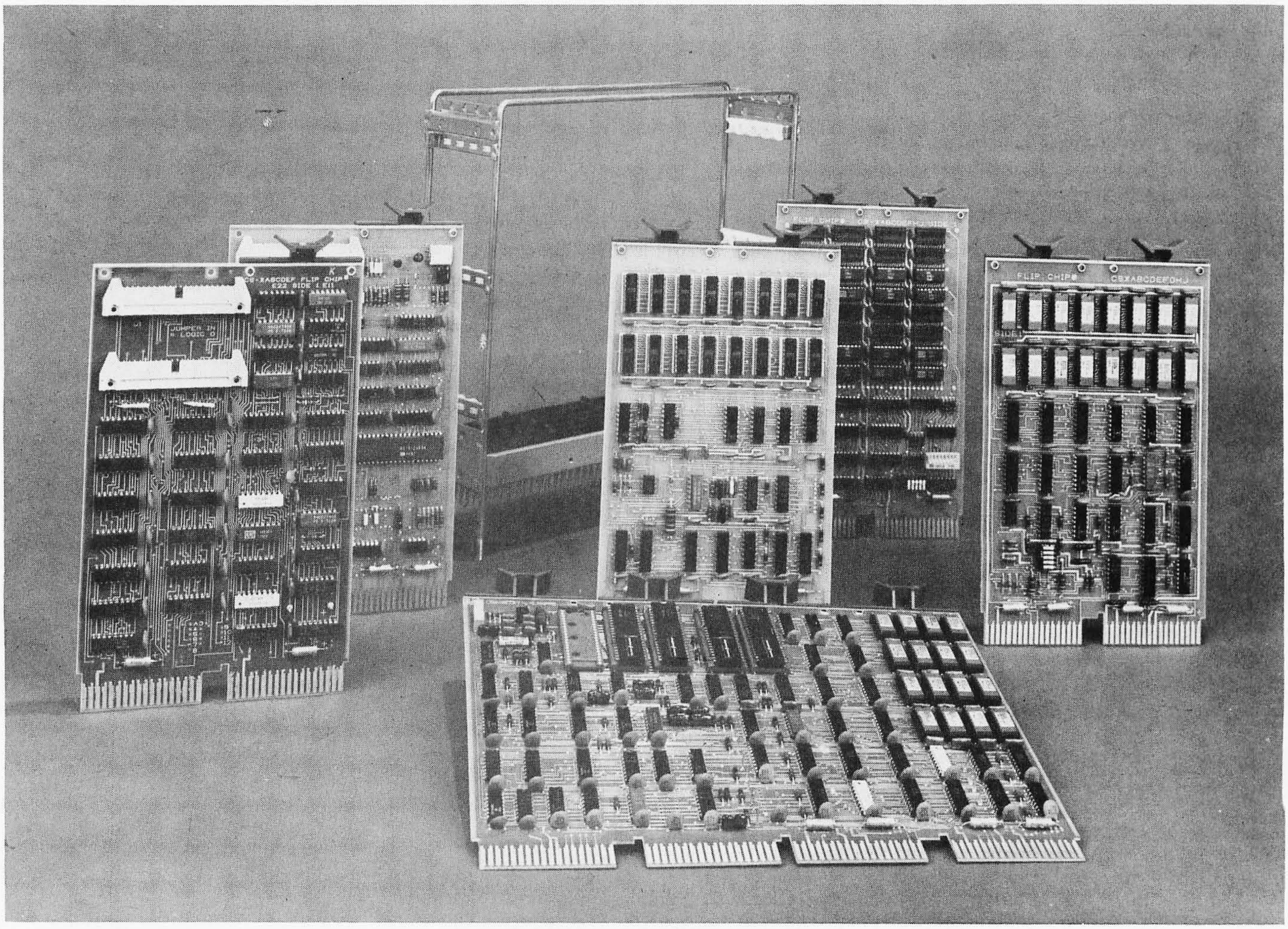
When enabled by an optional jumper, the CPU refresh control causes execution of a microcode subroutine approximately every 1.6 milliseconds; this operation refreshes all dynamic MOS memory in the system, not just the memory contained on the CPU module. While asserting a bus signal causing all dynamic memories to cycle at the same time, the CPU performs 64 memory references to refresh their contents. During the burst refresh time, external interrupts are locked out while DMA requests are still possible.

Maximum memory size of the 16 bit LSI-11 is 65,536 bytes or 32,768 words. Usually the top 4096 words of memory on members of the PDP-11 family are reserved for peripheral device control and data buffers, so the nominal maximum main memory size is 28,672 sixteen bit words. However, the user is not required to dedicate the entire upper 4096 word space to IO, but may implement only what is needed. Octal addresses 000 to 376 are usually reserved for trap and device interrupt vector locations. Several of these are reserved in particular for software generated interrupts (TRAPS) as shown in Appendix A.

Instruction Set

All operations are accomplished with one set of instructions rather than the conventional collection of memory reference instructions, operate/accumulator control instructions, and IO instructions. Single and double operand address instructions for words or bytes are used with a *wide range* of addressing modes, providing efficiency and flexibility in programming. The various addressing modes include sequential forward or reverse addressing, 8-bit byte addressing, 16-bit word addressing, and stack addressing. Using variable-length instruction formatting allows a minimum number of words to be used for each addressing mode.

Each processor instruction requires one or more bus cycles. The first operation fetches an instruction from the location specified by the program counter (PC). If no further operands are required for executing the instruction, no further bus cycles are used. If memory or an IO device is



RGS ELECTRONICS

DISCOUNTS: 10% OFF ORDERS OVER $25.00; 20% OFF ORDERS OVER $250.00.

SPECIAL
1-8008 **$50**
8-2102

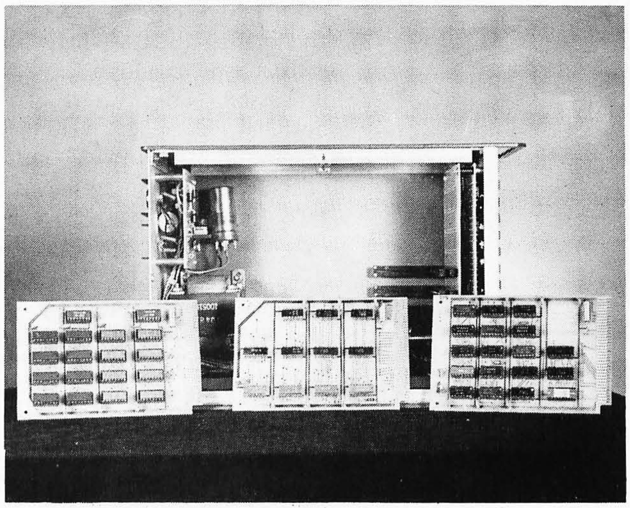
ANOTHER POWER SUPPLY . . .

PS 25–1 0 to 25v 1a lab type power supply with adjustable current limiting; remote sensing & remote programming for voltage & current. Insturctions included. All parts except chassis, meter(s), p.c. board.

Kit of parts with schematics. $14.95
P.C. boards available, No. 007 $3.00 ea.

2K RAM BOARD KIT. ALL PARTS INCL. SOCKETS

$84.50



ICs

8008 MICROCOMP. CHIP	$30.95
2102 1K STATIC RAM	3.00
5203 256x8 PROM	15.00
5204 512x8 PROM	25.00

INFO ON ABOVE CHIPS IF ASKED FOR.

008A MICROCOMPUTER KIT

8008 CPU, 1024 x 8 memory; memory is expandable. Kit includes manual with schematic, programming instructions and suggestions; all ICs and parts supplied except cabinet, fuses & hardware. Includes p. c. boards. $375.00

MANUAL ONLY, $25.00
(no discount on manual)

008A-K ASCII keyboard input kit. $135.00

008A-C Audio cassette adapter kit. $100.00

Details on computer, peripheral kits in our flyer.

ORDERS OF $50 OR MORE GET FREE BYTE SUBSCRIPTION IF ASKED FOR (CONTINENTAL U.S. ONLY).

RGS ELECTRONICS

3650 Charles St., Suite K ■ Santa Clara, CA 95050 ■ (408) 247-0158

We sell many ICs and components not listed in this ad. Send a stamp for our free flyer. TERMS OF SALE: All orders prepaid; we pay postage. $1.00 handling charge on orders under $10.00. California residents please include sales tax. Please include name, address and zip code on all orders and flyer requests. Prices subject to change without notice.

referenced, however, one or more additional bus cycles are required.

A special maintenance instruction is included in the LSI-11 instruction set to aid in hardware checkout. This instruction stores the contents of five internal registers in a specified block in main memory. A diagnostic program may then be used to examine the information and determine the internal operation of the micro-level processor.

The basic instruction set is that of the familiar DEC PDP-11/40 without memory mapping. Included are several operations normally not found even in other small PDP-11 processors, such as exclusive-or (XOR), sign extend (SXT), or subtract one and branch (SOB). There are also two new instructions used to explicitly access the processor status word (PSW). With the optional extended arithmetic chip, full integer multiply/divide and floating point arithmetic are also available. The instruction set is more comprehensive than that of the PDP-11/05 while the execution times are a little slower. Refer to Appendix B for a complete list of the LSI-11 instruction set and Appendix C for typical timings.

The branch instructions make use of the condition codes (PSW bits 0 to 3) which are set after execution of every arithmetic or logical instruction. This allows more efficient use of memory by eliminating extra instructions and temporary storage locations typically used to check results of various operations. The result of every operation is directly accessible and can be modified under software control by using any of the Condition Code Operator instructions. A list of the four condition codes along with a brief definition of each is listed in Appendix D.

Software

Since the LSI-11 uses standard PDP-11 software, there is an extensive library of programs available from DEC including diagnostic programs to check out your system after it is built. There is also a DEC Users Society (DECUS) which makes available a complete library of various PDP-11 programs at reasonable prices. Every LSI-11 owner automatically becomes a member of this organization.

ASCII Console

The conventional front panel lights and switches are replaced by an ASCII console/ODT package that operates with any standard terminal device communicating through a serial interface at a specific device address at any available baud rate. The functions available are very similar to those used by the familiar PDP-11 Octal Debugging Technique and are shown in detail in Appendix E. These include examining and changing the contents of memory and registers, calculation of effective addresses for relative and indirect addressing, and the functions of halt, single-step, continue and restart. By examining the contents of an internal CPU register, it is possible to determine which of the five methods of entering the console routines was used.

Upon entering the console routine, the location of the next instruction to be executed will be printed followed by @. The console routine will then wait for one of the 14 legal command characters. Thus, the user retains all the direct hardware control of a conventional lights and switches front panel and gains the ability to boot load from a specified device in byte transfer mode.

Interfaces

The LSI-11 system includes several standard interface modules to handle a variety of applications. Currently both a serial and a parallel IO interface is available, each as a single 8.5 inch by 5 inch (21.6 cm by 12.7 cm) PC board. The DLV-11 handles a single asynchronous serial line between 50 and 9600 baud, while the DRV-11 provides a full 16-bit parallel interface complete with two interrupt control units. The use of the two standard interface modules makes it very simple to connect any desired device to the LSI-11 bus. Standard devices such as teletypes, line printers, analog to digital converters, etc., can be connected directly to the interface modules with no additional circuitry. A simple cassette recorder interface can be made using the DRV-11 parallel interface, a UART chip, and a simple speed independent recorder interface circuit such as that shown in Don Lancaster's article *Serial Interface*, page 30, in the September issue of BYTE. ■

Are you interested in buying one?

This article has described the details of the LSI-11 computer by Digital Equipment Corporation. For those interested in purchasing the board version of this computer, the Southern California Computer Society is organizing a group purchase for amateurs. This purchase will involve an original equipment manufacturer (OEM) quantity of 50 or more machines, on a basis of cost plus 2% minimum contribution to SCCS. For further information contact Hal Lashlee of The Southern California Computer Society, at 213-682-3108. SCCS is organizing quantity purchases of other computer equipment, and is interested in making such offerings available through other computer clubs.

APPENDIX A: TRAP VECTORS

Location	Vector
ØØØ	(Reserved)
ØØ4	Time out & other errors
Ø1Ø	Illegal & reserved instructions
Ø14	BPT instructions
Ø2Ø	IOT instructions
Ø24	Power Fail
Ø3Ø	EMT instructions
Ø34	TRAP instructions
Ø6Ø	Console Input Device
Ø64	Console Output Device
1ØØ	External event line interrupt
244	FIS option

APPENDIX B:
LSI-11 INSTRUCTION SET

MNEMONIC	INSTRUCTION
Single Operand - General:	
CLR	Clear word
CLRB	Clear byte
COM(B)	Complement (1's)
INC(B)	Increment
DEC(B)	Decrement
NEG(B)	Negate (2's complement)
TST(B)	Test
Rotate & Shift:	
ROR(B)	Rotate right
ROL(B)	Rotate left
ASR(B)	Arithmetic shift right
ASL(B)	Arithmetic shift left
SWAB	Swap bytes
Multiple Precision:	
ADC(B)	Add carry
SBC(B)	Subtract carry
SXT	Sign extend
Processor Status (PSW) Operators:	
MFPS	Move byte from PSW
MTPS	Move byte to PSW
Double Operand - General:	
MOV(B)	Move
CMP(B)	Compare
ADD	Add
SUB	Subtract
Logical:	
BIT(B)	Bit test (logical AND)
BIC(B)	Bit clear
BIS(B)	Bit set (logical OR)
XOR	Exclusive OR
Branches:	
BR	Unconditional branch
BNE	Branch if not equal to Ø
BEQ	Branch if equal to Ø
BPL	Branch if plus
BMI	Branch if minus
BVC	Branch if overflow is clear
BVS	Branch if overflow is set
BCC	Branch if carry is clear
BCS	Branch if carry is set
Signed Conditional Branches:	
BGE	Branch if greater or equal to Ø
BLT	Branch if less than Ø
BGT	Branch if greater than Ø
BLE	Branch if less or equal to Ø
Unsigned Conditional Branches:	
BHI	Branch if higher
BLOS	Branch if lower or same
BHIS	Branch if higher or same
BLO	Branch if lower
Condition Code Operators:	
CLC	Clear C Condition Code Bit
CLV	Clear V
CLZ	Clear Z
CLN	Clear N
CCC	Clear all condition code bits
SEC	Set C Condition Code Bit
SEV	Set V
SEZ	Set Z
SEN	Set N
SCC	Set all condition code bits
Jump & Subroutines:	
JMP	Jump
JSR	Jump to subroutine
RTS	Return from subroutine
MARK	Mark (aid in subroutine return)
SOB	Subtract 1 & branch if not Ø
Trap & Interrupts:	
EMT	Emulator trap
TRAP	Trap
BPT	Breakpoint trap
IOT	Input/Output trap
RTI	Return from interrupt
RTT	Return from interrupt
Miscellaneous Instructions:	
HALT	Halt
WAIT	Wait for interrupt
RESET	Reset external bus
NOP	(no operation)
Optional EIS:	
MUL	Multiply
DIV	Divide
ASH	Shift arithmetically
ASHC	Arithmetic shift combined
Optional FIS:	
FADD	Floating add
FSUB	Floating subtract
FMUL	Floating multiply
FDIV	Floating divide

Continued on page 22

VIDEO DISPLAY MODULE



Corporation

2465 Fourth Street
Berkeley, Ca. 94710

Continued from page 19

APPENDIX C:
TYPICAL INSTRUCTION TIMING

INSTRUCTION	TIME (usec)	COMMENTS
ADD R1,R2	3.5	Register addressing
MOV R3,RØ	3.5	
MOV TAG1,1Ø (R2)	11.55	Relative & index addressing
TSTB (R3)+	5.25	Auto-indexed
BMI TAG2	3.5	Conditional branch
JSR PC,2(R2)	8.Ø5	Subroutine call
JMP (R4)	4.2	Jump indirect
RTI	1Ø.5	Return from interrupt
Optional EIS & FIS Instructions:		
MUL	24 - 64	Multiply
FADD	42.1	Floating add
FMUL	52.2 - 93.7	Floating mult
FDIV	151 - 232	Floating divide

APPENDIX D:
PSW CONDITION CODES

CODE	PSW BIT	CONDITION WHEN SET = 1
N	3	If result were negative
Z	2	If result were zero
V	1	If operation resulted in an arithmetic overflow
C	Ø	If operation resulted in a carry from the msb (most significant bit) or a 1 was shifted from the lsb (least significant bit)

APPENDIX E:
ASCII CONSOLE/ODT COMMANDS

Command	Function
<CR>	Close opened location and accept next command.
<LF>	Close current location; open next sequential location.
Up-arrow	Open previous location.
Back-arrow	Take contents of opened location as a relative address, and open that location.
@	Take contents of opened location as absolute address and open that location.
r/	Open word at location r.
/	Reopen the last location.
$n/ or Rn/	Open general register n(Ø-7) or S (PSW).
r;G or rG	Go to location r and start program
nL	Execute bootstrap loader using n as device CSR. Console device is 17756Ø.
;P or P	Proceed with program execution.
RUBOUT <DEL>	Erases previous numeric character. Response is a backslash (\).

Continued on page 24

Go Computer Now!

Why not?

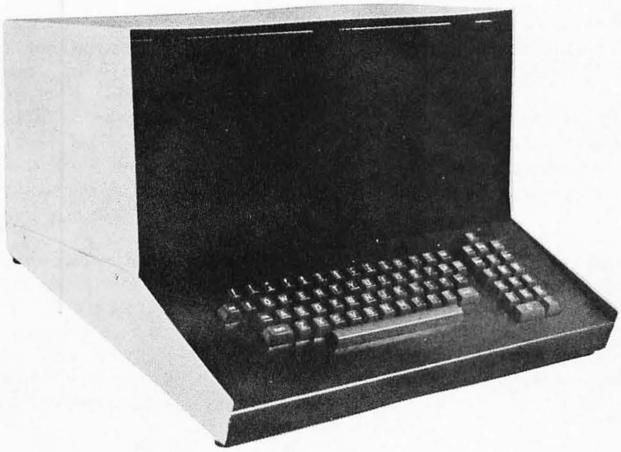
FROM $860 TO $11,300 SPHERE CAN'T BE BEAT!

SPHERE starts with a CPU using a Motorola 6800 microprocessor, a Real-Time Clock, 4K of dynamic memory, 1K of PROM software. The CRT Board generates 16 lines by 32 characters of ASCII on a television or video monitor. Keyboard is complete with numeric and cursor editing keypads. From here, hardware can be expanded to your hearts desire with extra memory boards (up to 64K), serial communications interface, cassette interface, Modem, digital I/O (as many as you need). Floppy Disk memory (up to 4 disks), 8 computer terminals, line printer etc...all from one M6800 chip.

With a SPHERE Computer, stand-alone development is just the beginning, you can configure your system to handle your problem solving/record keeping needs. All SPHERE Computer Systems come complete with useable software languages. Available are "PDS" 1K Basic, or extended Basic Compiler. When computer is turned on, it immediately goes into a command mode, so that you can instantly start programming. "PDS" contains a mini-assembler, editor, debugger, and utility command set in 1K of PROM. Also available is a 1K subset of Basic. Our extended Basic compiler is complete with string, matrix, and file functions, and requires 12K of memory. With this software you can perform your applications whether it be accounting, home management, education, security monitoring, research, business, etc. Why not invent your own application? For play or for work your biggest problem is no problem at all. **$860 is the start for an operating Computer System Kit.** Your computer is ready and comes complete with operator manuals sufficient for first-time computer users. Contact us today for more information.

KIT	ASM	
$350	$520	**ONE-CARD COMPUTER:** Motorola 6800 microprocessor, 4K RAM, 512 bytes EPROM (containing a Program Development System), a REAL-TIME CLOCK, 16 LINES OF DIGITAL I/O, hard wired ROM Monitor, and a serial type interface. This is the 100-quantity price, extended to the hobby user for a limited time on a single unit.
522	622	**CPU BOARD:** Motorola 6800 microprocessor, 4K RAM, 1K EPROM (containing an EDITOR, ASSEMBLER, DEBUGGER, COMMAND LANGUAGE, CASSETTE LOADER, DUMPER, UTILITIES), and a REAL-TIME CLOCK.
860	1400*	**SPHERE 1:** Includes the CPU BOARD described above, plus 512 character video with full ASCII keyboard and numeric/cursor keypad, power supply, chassis, manuals and associated parts.
$999	$1499*	**SPHERE 2:** Includes all features of SPHERE 1, plus serial communications and audio cassette or MODEM interface.
1765	2250*	**SPHERE 3:** Includes all the features of SPHERE 2, plus memory totaling 20K which is sufficient to run full extended BASIC Language.
6100	7995*	**SPHERE 4:** Includes all of the features of SPHERE 3, except the cassette has been replaced by an IBM-compatable Dual Floppy Disk System. This system includes a Disk-operating System and BASIC Language and a 65 LPM line printer.
(various)		**OTHER SPHERE PRODUCTS:** Light pen option; full color and B/W video graphics system; low cost Dual Floppy Disk System; and full line of low cost peripherals.

*This ASSEMBLED SPHERE System includes the complete chassis, and video monitor as pictured below.



SPHERE CORPORATION

791 South 500 West Dept. 129 (801) 292-8466
Bountiful, Utah 84010

Continued from page 22

BUS — A collection of parallel data paths and power lines used to interconnect the various elements of the system, including the central processor, memory, and all peripherals.

BUS TIMEOUT — Bus timeouts or bus errors occur whenever the controlling device on the bus (the bus Master) does not receive a response from the addressed device (the Slave) within a certain length of time. In general, these are caused by attempts to reference non-existent memory or peripheral devices. Bus error traps cause processor traps through the trap vector address 4.

DMA — Direct Memory Access. For high speed devices, memory may be accessed directly through the bus without the use of program controlled data transfers.

JUMPER SELECTABLE OPTIONS — The CPU module contains locations for six wire jumpers to control the various operating options as follows:

1. Two wires select which of the four possible power-up options is desired. These are normally set to restart through vector location 24 (so the LSI-11 acts as a standard PDP-11).

2. One wire jumper enables the external event (or real-time clock) interrupt feature when inserted.

3. One wire jumper enables the automatic dynamic memory refresh feature when inserted.

4. Two wire jumpers determine the addressing of the 4K RAM memory located physically on the CPU module.

Each wire jumper consists of a short length of bare copper wire soldered between two designated holes in the PC board.

MACROCODE — The instruction set which the programmer sees and actually uses to implement his program, such as the PDP-11 instruction set in this case.

MICROCODE — The low level instruction set used in a microprogrammed processor to "emulate," or execute, the macrocode. Microcode is more primitive in function, but executes at a higher speed than the macrocode.

PC — Program Counter. A register which contains the address of the next instruction to be executed.

POWER-UP SEQUENCE — Two wire jumpers on the CPU module select one of the four possible power-up modes:

OPTION	POWER-UP
0	PC at 24, PSW at 26, or HALT
1	ODT – ASCII Console
2	PC = 173000, or HALT
3	Special processor microcode

The power-up sequence is initiated upon supplying power to the processor module or on restoration of power after a temporary power fail has occurred.

PSW — Processor Status Word. Contains information on the current status of the processor including the priority mask (bit 7) and the condition codes (bits 0 to 3).

STACK — An area of memory set aside by the programmer for temporary storage or subroutine/interrupt sevice linkage. The stack uses the "Last In – First Out" concept; thus various items may be added to a stack in sequential order and retrieved or deleted from the stack in reverse order. Stack starts at the highest location reserved for it and expands linearly downward. In the LSI-11, register 6 is reserved as the hardware stack pointer and must be initialized by the software. However, registers 0 to 5 may be used for various program defined stacks as needed.

TRAP — Software generated interrupt.

VECTOR — A unique address which points to a reserved set of locations (2 words) for interrupt or error handling. The first word contains the starting address of a service routine (a new PC) while the second holds the new PSW to be used by the service routine.

VOLATILE MEMORY — Volatile memory, such as RAM, will not retain useful information without power applied continuously. Non-volatile memory, such as core or ROM, will always retain its information with or without power applied.

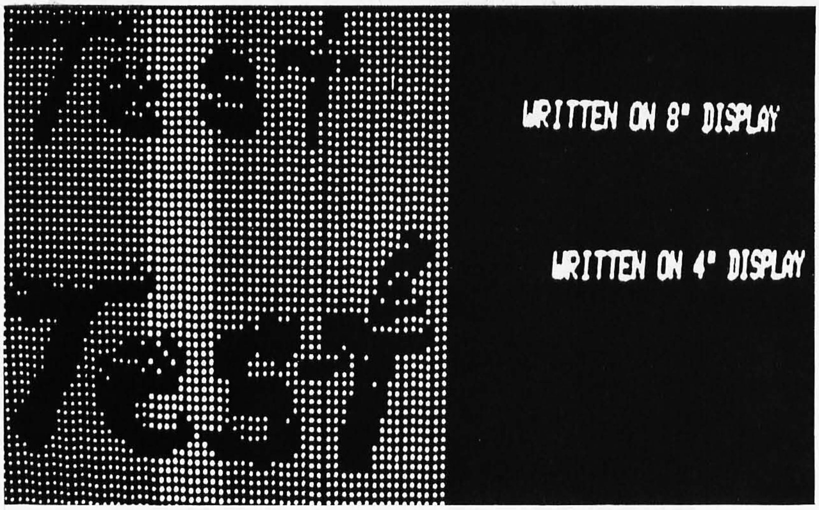


Photo 1: The ease of removing just one dot from a full field display depends upon the display size. The author's X-Y display has an adjustable size control which was used in preparing this picture.

Let There Be Light Pens

Sumner S. Loomis
Loomis Laboratories
Route 1 Box 131A
Prairie Point MS 39353

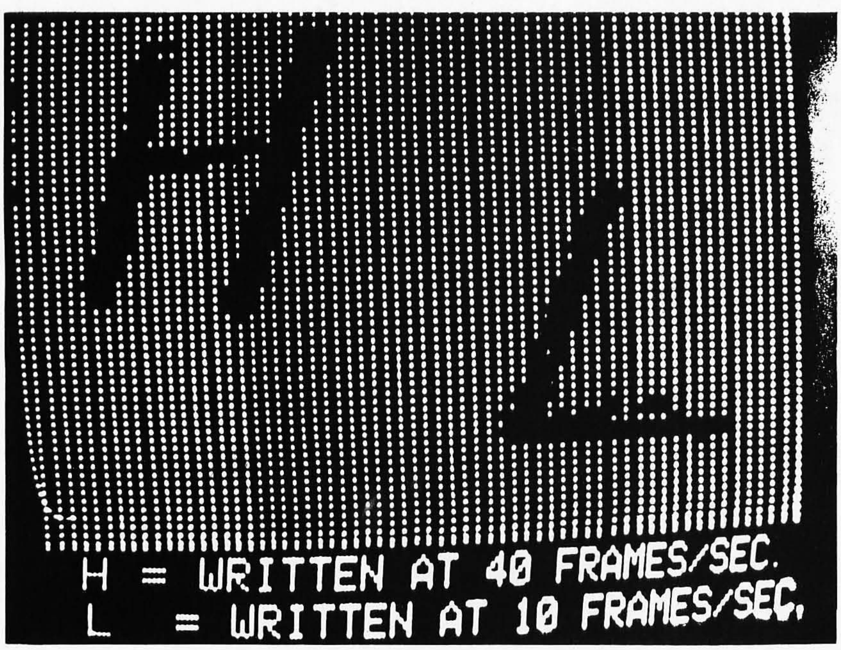


Photo 2: The light pen can be used in an "erase" mode by filling the screen with "on" dots then selectively removing dots with the light pen. The titles added to this picture (and all the pictures in this article) were created with a separate character generator which is not described.

With only a few components and a few hours of construction you can add a versatile light pen to the oscilloscope graphics interface which has been described in the October 1975 issue of BYTE, page 70 ff.

By holding the light pen to the face of the cathode ray tube (CRT), a point may be added or removed. This eliminates the awkward and time consuming effort required when using a program or manual switches to change the dots on the screen.

The resolution and capability of the light pen are dependent on two characteristics of the CRT. The brightness and the size of the display tube will determine how easily you may add or remove one dot. An idea of the effect of display size may be had from photo 1. The word Test was written twice on a 12 inch (30.5 cm) black and white TV picture tube configured as an XY display like an oscilloscope. The top word was written with the display adjusted to an 8 inch (20.3 cm) size, and the lower word was written with a 4 inch (10.2 cm) display. Each letter was written with only one stroke of the light pen without touch up or corrections. With some practice, and possibly several passes, one dot may be added or removed if the display measures 8 inches (20.3 cm) or more. Further improvements to the pen are required with smaller display tubes. An advanced circuit that greatly improves the capability of the pen with small displays is also described in this article.

The light pen can erase or draw depending on the setting of a switch. Examples of the two actions may be seen in photos 2 and 3. If the oscilloscope interface is adjusted for a high repetition rate, some smearing or carry over into the neighboring dot positions occurs. The author's system has a front panel control permitting ten repetition rates. A small improvement in resolution can be noted at the lower writing rates, as shown in photo 2. A frame consists of 64 by 64 dots.

Theory of Operation

The light pen operates on the principle that brightness is quite intense during the actual interval that a particular dot is being written by the CRT's electron beam. Although phosphor will continue to emit light for some time, the brightness decays in an exponential manner after the writing beam has moved on to the next dot.

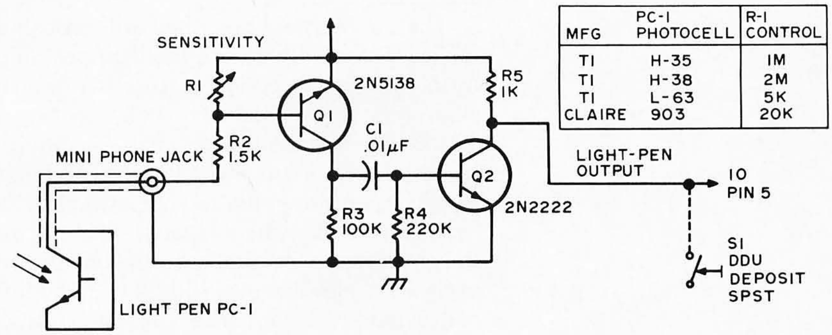
Figure 1 illustrates the simple light pen circuit. With proper adjustment of the sensitivity control (and possibly the brightness control), the photocell in the tip of the light pen will sense the moment in time when a dot is written at the particular location of the light pen. At this instant, the photocell will conduct, biasing the PNP transistor which causes a short pulse to be conducted through capacitor C1 to the base of the NPN transistor Q2. If the pulse is greater than .6 V, this transistor will be driven into saturation, and the light pen output will fall to .3 V. This output line is the connected to pin 5 of the oscilloscope graphics unit which writes a 1 or a 0 bit (dot or no dot) at precisely the instant that the dot position touched by the pen was addressed.

The above procedure works quite well if the dot to be changed is illuminated at the time. With proper adjustment of the sensitivity control, we can usually use an illuminated dot just above the point of action (it must precede the dot in scan sequence) to create a new dot in the next space. This action of extending a line can be quite useful for drawing bar graphs on the CRT. This mode of entry is possible because screen persistence allows the light pulse to be carried over into two or three subsequent dot positions depending on the frame speed.

How can the photocell sense the dot's position if there is not any illumination to trigger it? This is accomplished by the flood circuit which is shown in figure 2. This circuit overrides the normal Z-axis control and floods the screen with light by feeding a logical one signal to the Z axis of the display unit. With this arrangement the pen is placed at the required dot position, the footswitch is actuated to flood the screen with light, and the photocell is energized when the



Photo 3: The light pen can be used in an "enhance" mode by using a footswitch control to flood the screen momentarily when the light pen is in position.



MFG	PC-1 PHOTOCELL	R-1 CONTROL
TI	H-35	1M
TI	H-38	2M
TI	L-63	5K
CLAIRE	903	20K

Figure 1: The simple version of the light pen can be constructed according to this schematic. All resistors ¼ W.

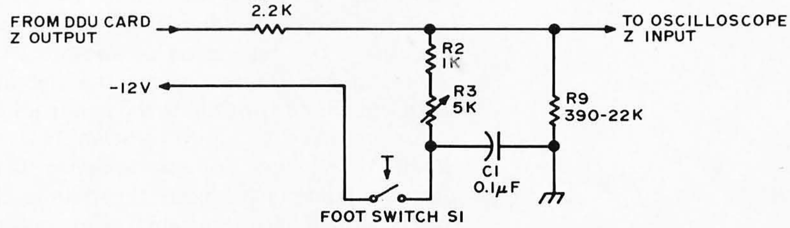


Figure 2: The foot switch control used to flood the screen for "enhance" mode operation is given in this circuit which modifies the Z-axis signal to the oscilloscope driver.

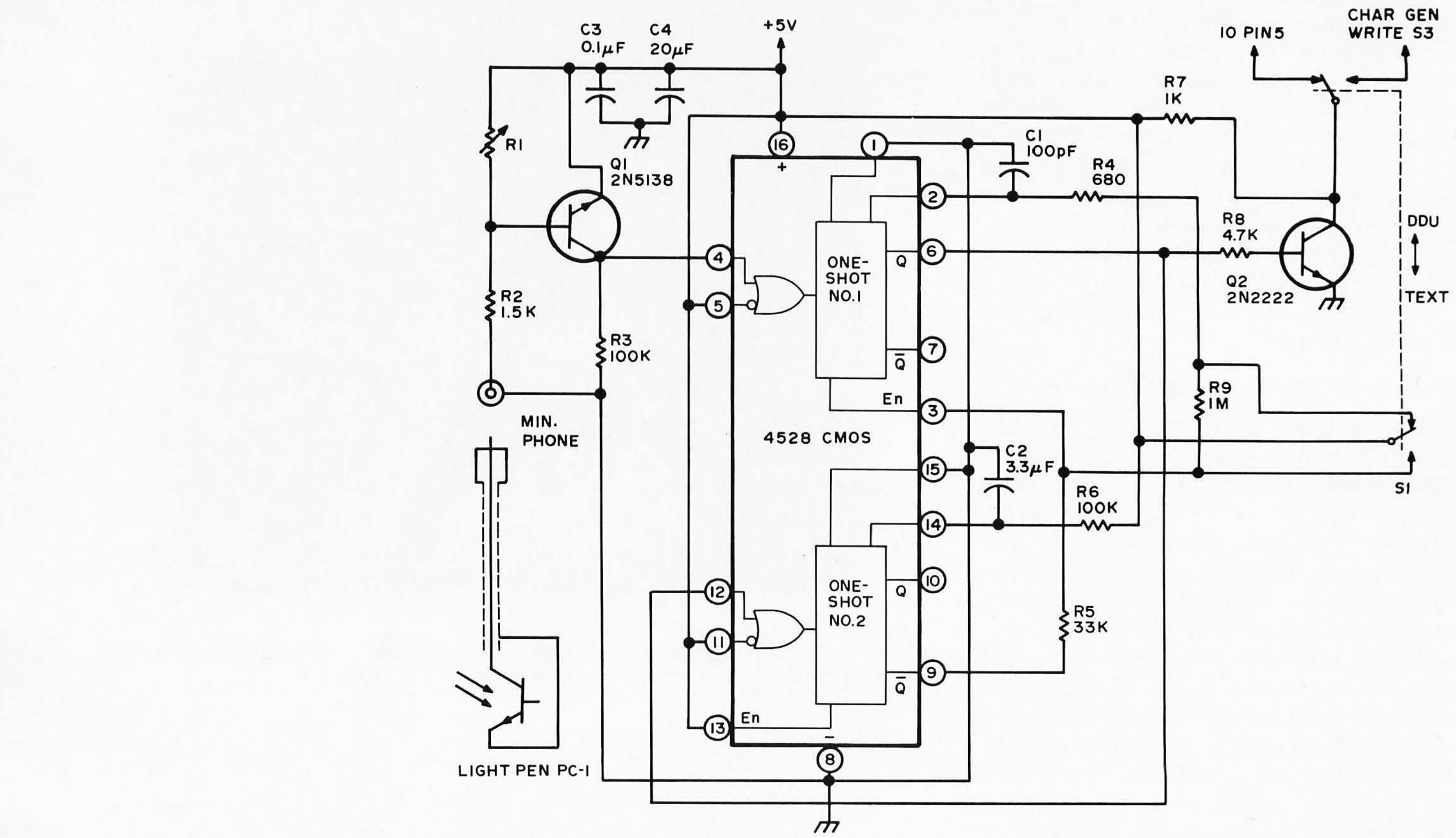


Figure 3: By adding a pair of oneshots to the circuit, the ability to draw pictures is improved through a short data lockout period which avoids smearing.

writing beam reaches that particular dot position. Releasing the footswitch removes the flood and allows the data to be examined.

The circuits just described will probably suffice if you wish to use the light pen only for occasional correction of data. If you plan extensive and detailed work, such as cartooning or statistical data entry, a modification of the circuit will allow you to tailor the light pen's response to your own particular needs and system speed. The circuit shown in figure 3 is similar to the one shown in figure 1. However, it includes two oneshot multivibrators (contained in one CMOS DIP). The first one produces a constant amplitude pulse of approximately 200 nsec duration which is sufficient to bring about the storage of a 1 or 0 bit in most versions of the 2102 memory (ICs 11 to 14 in the oscilloscope graphics interface). The second one delays the generation of another write command for .25 sec, giving the operator sufficient time to withdraw the pen from the screen or move to a new location, before a double or multiple dot can be drawn. Once the two pulses have been timed in accordance with a given system speed and the operator's writing speed, it becomes very easy to draw detailed images with the light pen.

In figure 3 resistor R4 and capacitor C1 control the length of the write pulse, and resistor R5 and capacitor C2 control the wait time. For the 4528 CMOS oneshot, the time of the pulse (T) measured in microseconds is a function of resistance (R) and capacitance (C) measured in ohms and microfarads, respectively, as follows:

T = 2.5 * R * C ** .85;

where a single asterisk denotes multiplication, and a double asterisk exponentiation.

The circuit shown in figure 3 also includes a switch and connections for using the light pen with the author's text display and editing system. Exact details for this connection are not given here, as they will differ with the type and construction of the text display system. I found, however, that the shift register type memories commonly used in these systems require a much longer write pulse than is necessary for the 2102 memories. It was also desired to eliminate the holdoff circuit (second oneshot) for this application. These changes are accomplished with switch S1 and resistors R9 and R5. If these features are not desired, it is recommended that R9 be replaced by a wire, R4 changed to 4.7 kΩ and C1 to 20 pF.

Construction

As is shown in the table accompanying figure 1, several different types of photocells are suitable for use in the light pen. The Texas Instrument (TI) type H-35 or H-38 is a very small device with a built in lens. These were originally designed for use in punched tape and card readers, thus the small size. Their size, sensitivity, and restricted field of view make them ideal for this application. The high impedance of these devices, however, makes them somewhat slow for this application, particularly at low brightness levels. The slow response time limits their use at the faster scan rates, and complicates the smearing mentioned earlier. Another device, the L-63 type which is available from Radio Shack (276-140 infared detector), was found to be considerably faster. Being a much larger device, however, it has a larger field of view, and much of its speed advantage is lost to optical smearing. Models of both photocell types were built and tested by me, with only slight preference for the H-35. With some careful masking, and possibly the addition of a small, short focal length lens (e.g., Edmund Scientific number 12050 cylinder lens, or a small drop of clear epoxy), this photocell will probably perform better than the H-35 for this application. The Claire types 903 and 903-L were tried with only fair results.

Any ball-point pen or felt-tipped marker can be reworked to make a housing for your light pen. Take a tour of the local stationery store to find likely candidates. The L-63 photocell was found to fit nicely into the end of a Graphi-100 marker pen which can easily be disassembled with diagonal cutters. An example of the construction with the L-63 is shown in photo 4, and the H-35 assembly is shown in photo 5.

Secure the photocell in place with epoxy adhesive after attaching the shielded cable. The cable can also be secured against damage from pulling by filling the entire pen with silicone rubber adhesive or ordinary household bathtub caulk. It is wise to keep the cable short, especially with the H-35 or H-38 photocells, to obtain maximum possible response speed. I used an 18 inch (45.7 cm) long miniature coaxial cable leading to a miniature phone plug.

If you are using the simple circuit of figure 1, the parts can be assembled on a small turret terminal board available at most electronic supply houses. This assembly is shown in photo 6. The circuit of figure 3 can be assembled in the same manner with the addition of a 16 pin DIP socket. R4, R5, C1 and C2 should be mounted in such a manner that they can be changed easily (Cambion 601-1512 component clips are useful here).

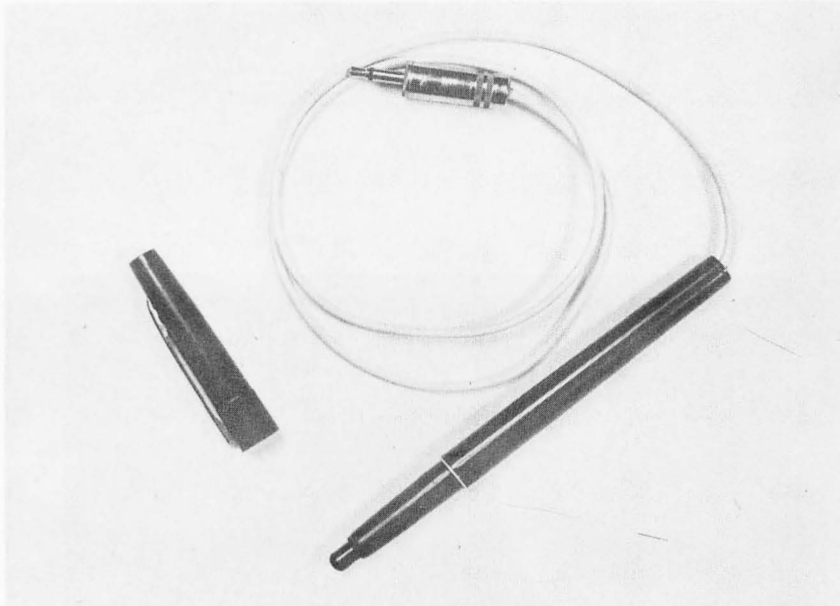


Photo 4: This shows a pen based on the TI type L-63 photocell, built using a marking pen case.

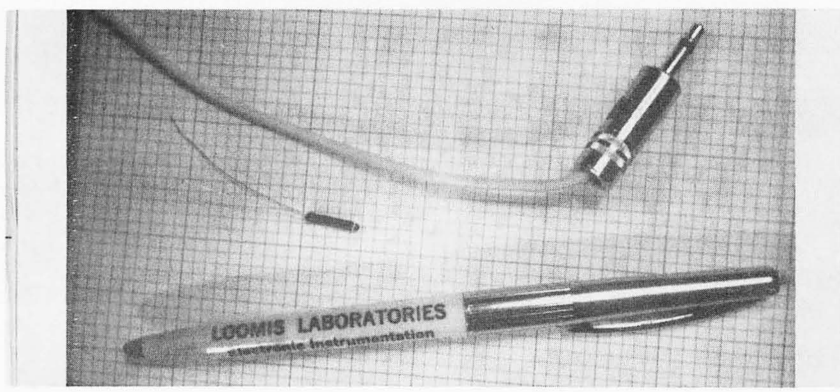


Photo 5: This picture shows an assembled light pen using a TI type H-35 (or H-38) photocell with a standard ballpoint pen housing.

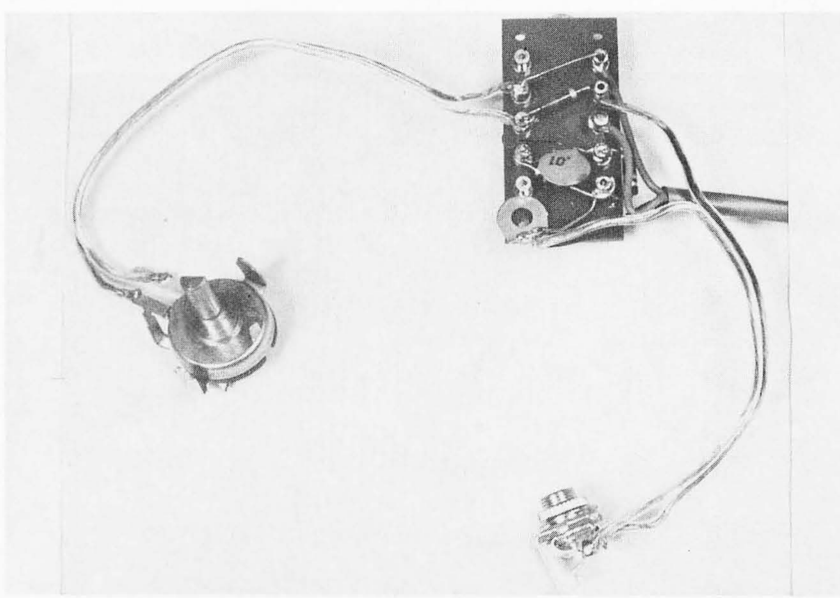


Photo 6: This photo illustrates how the circuit of figure 1 can be assembled using a small turret terminal board. The transistors and R5 are mounted out of sight on the rear side of the board.



Photo 7: Using the improved circuit of figure 3 reduces much of the over-writing of multiple dots which occurred using the original circuit of figure 1. This is an enhance mode picture.

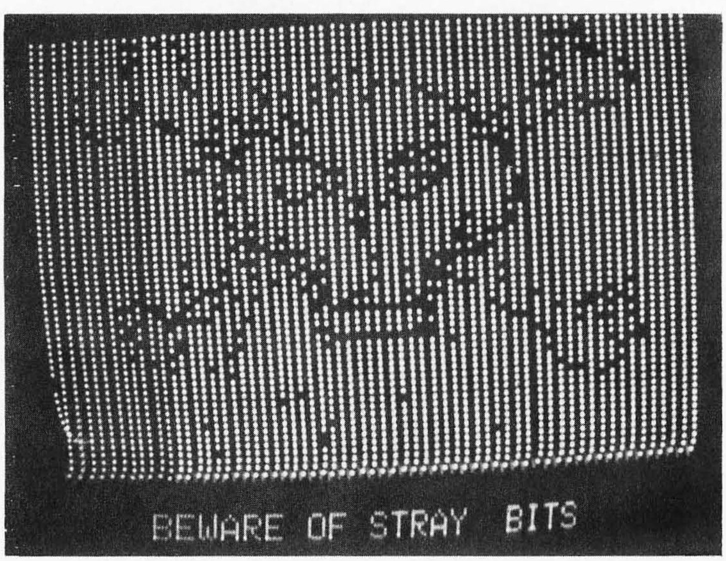


Photo 8: This illustrates a cartoon drawn using the erase mode of operation with the improved circuit of figure 3.

The sensitivity control may be conveniently mounted on the front panel.

The operation of the light pen requires control of the inputs to the oscilloscope graphics unit. I have found that one of the most convenient ways in my system is through a set of manual data switches. This type of input was illustrated as a test fixture for the oscilloscope graphics interface in figure 5 on page 75 of the October 1975 issue of BYTE. In my system, these data input switches are shared with a Mark-8 minicomputer front panel by means of an 8 pole double throw toggle switch. It is also possible to set up input codes to the oscilloscope graphics unit using software in the microcomputer system which drives it.

In order to enter data with the light pen, a deposit switch is pressed whenever the pen is in the proper position for data entry. The deposit switch should be mounted in a convenient location near the display tube and light pen. In my system the light pen deposit switch was mounted next to the original deposit switch of the Mark-8 computer.

Using The Light Pen

To illustrate the use of the light pen, we will cover the procedure necessary to draw a simple figure on the screen in the erase mode using manual controls. Set the switch register to 1000 0110 binary (turn scan on) and depress the deposit switch. This should produce random dots on the screen. Set the switch register for 1000 0010 binary (set Z on) and depress the deposit switch again. The screen will show a full field of dots. (If the Z axis polarity of your display tube is reversed, you will have to use the "set Z off" command (1000 0011 binary) to illuminate the screen.) Set the switch register for 1000 0010 (set Z on), but *do not* activate the deposit switch. Now bring the light pen in contact with the display CRT, and note that the dot or dots within its field of view are erased. To erase the entire screen and start over, simply press the deposit switch and repeat the above procedure.

To write in the enhance mode (screen dark, writing illuminated dots), reverse the above procedure by wiping the screen clean with the "set Z off" command (while the scan is on), and after setting the switch register to "set Z on" without the deposit switch, proceed to write dots with the light pen. In this mode, the flood foot switch must be periodically activated to provide the required illumination. Examples of the light pen's drawing capability can be seen in photos 7 and 8. ■

Horror Story

Jim C. Warren Jr.
Star Route Box 111
Redwood City CA 94062

Not too long ago, researchers at Stanford Medical Center in California were horrified to discover that several years of data that were stored on magnetic tape had disappeared. The tapes hadn't disappeared, just the data. The discovery was made when they attempted to retrieve some of the data for analysis, but found only "garbage" recorded on the tapes. Even more disturbing was the fact that these tapes were supposed to be ultra reliable. They had been especially developed for storage of important research data and used a fully redundant recording technique for improved reliability.

Fully redundant recording is a storage technique in which each data bit is recorded in two different locations. In this case, DECtape™ was being used. This is tape that is 3/4ths of an inch wide and has six parallel data recording channels or paths. The data recorded in the three channels on one side of the tape is duplicated in the three channels on the other side. Thus, the tapes have half the data capacity that they could have, but their reliability is significantly improved.

It should be noted that there are a number of other possible redundant storage techniques in use for improving reliability. For instance, "triply redundant recording" replicates each bit three times in three separate locations. Alternatively, depending on the allowable bit-patterns that may be used to store data, "partial redundancy" may be used. In this technique, a computation is performed on the explicit bit-pattern of each datum. The computation result requires fewer bits than the original datum, but may be used to either replicate the original datum (thereby recovering it, if necessary) or to at least verify the authenticity of the original datum. These are referred to as "error-correcting codes" and "error-checking codes," and form an on-going area of applied-mathematics research. In either case, partial redundancy is important in that it improves reliability, but requires less additional storage than full- or triple-redundancy.

First, the researchers thought (prayed) that their tape decks were malfunctioning. Not so. Diagnostic checks were made and the tape equipment was working properly. Next, the tapes were examined. Nothing was wrong with them, physically, and in fact, new data could be recorded on them and retrieved without difficulty. They asked others around the Medical Center if they had encountered similar problems (there are about 50 computers at the Stanford Medical Center). None had. Then they looked for environmental causes, but there appeared to be none. The temperature and humidity recorders, common to biomedical research facilities, indicated no significant fluctuations. There had been no fires, no chemical accidents, and there was no x-ray or high power electronics gear in use nearby. The tapes had been in their individual boxes, just like the tapes for all of the other computer facilities in the Medical Center, and these boxes had been neatly stored on the bottom shelves of a cabinet, well out of the way of possible harm. It was a most frustrating puzzle.

Finally, however, the mystery was solved. It seems that the janitor had made his biannual floor polishing rounds, using a heavy duty rotary floor polisher. The magnetic radiation from its massive motor, in proximity to the low shelved tapes, had raised havoc with the bit patterns that had been recorded on the tapes.

The researchers now store their tapes on the top shelves. ■

LIFE Line 4

Integrating graphics control commands

Carl Helmers

In LIFE Line 3, the design of the DECODE routine of the LIFE program was presented. DECODE is designed as a table driven mechanism for selecting one of several subroutines which carry out the functions of the LIFE program's KEYBOARD_INTERPRETER. However if you examine table 1 of LIFE Line 3 (see p. 51 of BYTE #4), you will note one conspicuous and intentional lack: There are no routines which process the interactive graphics commands required to set up LIFE patterns on the scope display. Yet in LIFE Line 1, several special purpose keys were introduced as manual inputs for cursor motion control and data definition purposes. Where is the missing part of the program which interfaces these keys? What are the hardware implications of requiring a special keyboard? Answers to these questions are the major concern of LIFE Line 4. Integrating the graphics control commands is a combined hardware and software topic. The software is that of the DEFAULT routine that interprets several keyboard inputs not handled by DECODE; the hardware consists of the design of a special keyboard interface to automatically switch between an ASCII keyboard's 7-bit parallel output code and the LIFE graphics control keypad.

The main requirement for LIFE cursor motion and data control is that one, two or three of the input keys can be depressed at the same time. This capability is needed in order to specify all the possible combinations of motion control and optional cell birth or death data inputs. The individual motion control possibilities (one key at a time) are the movements in four principal directions: up, down, left or right. When two motion control keys for perpendicular directions are selected at the same time, diagonal motion is the desired result. With either form of motion control, entry of data can optionally be performed by depressing either the birth key or the death key at the same time. Thus as many as three keys may be sensibly pressed simultaneously when entering data.

The large number of combinations possible for the six bits which will be needed for six switches strongly argues against making the software use a table driven algorithm such as DECODE. This is the reason why no cursor motion and data entry commands are found in table 1 of LIFE Line 3. Since each bit of the parallel information from the motion control switches can have an independent meaning, a specially programmed determination of motion control actions uses less memory than the huge table which would be required for all the combinations. Thus handling of motion control is left to the DEFAULT routine which is called by DECODE when it fails to decode one of the commands in table 1 of LIFE Line 3. (DEFAULT also handles ASCII numeric inputs, as you'll see a bit later in LIFE Line 4.)

Graphics Control Hardware Considerations

For the hardware of LIFE, how can the need of this special set of input codes be reconciled with the need to input ASCII via the same eight-bit input port? One answer lies in the choice of an eight-bit format in which the most significant bit determines what lies in the low order seven bits. With this format, one state of the most significant bit indicates when an ASCII code is present in the low order; the other state of the most significant bit indicates when graphic control keyboard information is in the low order. This choice of format is supported in hardware by the addition of a simple interface module which uses seven integrated circuits to switch between data sources and debounce the motion control keyboard.

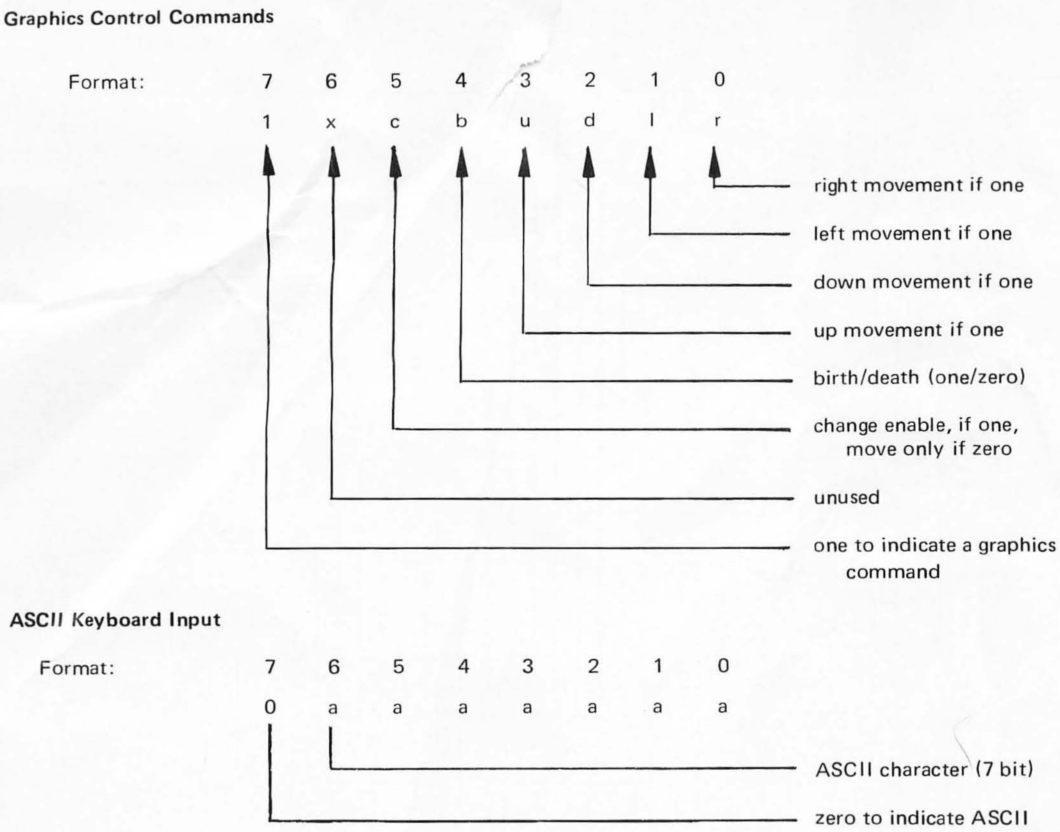


Figure 1: Data formats for graphic control commands and ASCII keyboard input.

The combined ASCII and control data format is illustrated in figure 1. When the value of bit seven of the interface is read as one, the programming of the DEFAULT routine will always be entered and the low order bits will be analyzed as graphic control information as shown by the upper diagram in figure 1. The low order bits zero through three represent the individual key states of the motion control switches and the next two bits, four and five, are encoded with information on data entry from the birth and death switches. If the value of bit seven is read as a logical zero, the program will interpret the ASCII value of the low order bits through the DECODE routine of LIFE Line 3, or through the DEFAULT routine if the command is not in the table which drives DECODE.

The hardware needed to implement this special interface is shown in figure 2. The interface consists of a two way data selector (IC6 and IC7) which determines whether the eight bit pattern presented to the system bus interface comes from the cursor motion control keyboard or from the ASCII keyboard. The ASCII data is routed straight to the data selectors from a jack (J1) which receives a cable which connects to the keyboard unit. (The LIFE Line system prototype is currently using one of the surplus Sanders 720 keyboards described in BYTE #1.) The graphic control information is derived through jack J2 from the special keyboard via the 7474 flip flops IC1, IC2 and IC3. These D flip flops are being used as set reset flip flops by grounding the clock line and employing the preset (PRES) and clear (CLR) inputs for data and keyboard acknowledge functions respectively. The flip flop outputs for bits zero to three go directly to the data selector to define cursor motion inputs. The flip flop output for bit four (birth switch) also is directly connected to the selector. However, bit five of the selector's cursor motion inputs is taken from NAND gate IC4D which encodes a CHANGE ENABLE signal when either birth or death data input is indicated. (Note that the user of the LIFE cursor motion control keyboard is on his honor not to push both birth and death keys simultaneously – with this encoding logic, birth always locks out death.) One item derived from the cursor control keyboard is a key pressed signal produced by 7430 NAND gate IC5. This signal is inverted by IC4C and used to control the data selector: If any key on the cursor motion control keyboard is pressed, the ASCII keyboard will be locked out;

A flip flop with preset and clear inputs can be used in place of a hand-wired set reset flip flop.

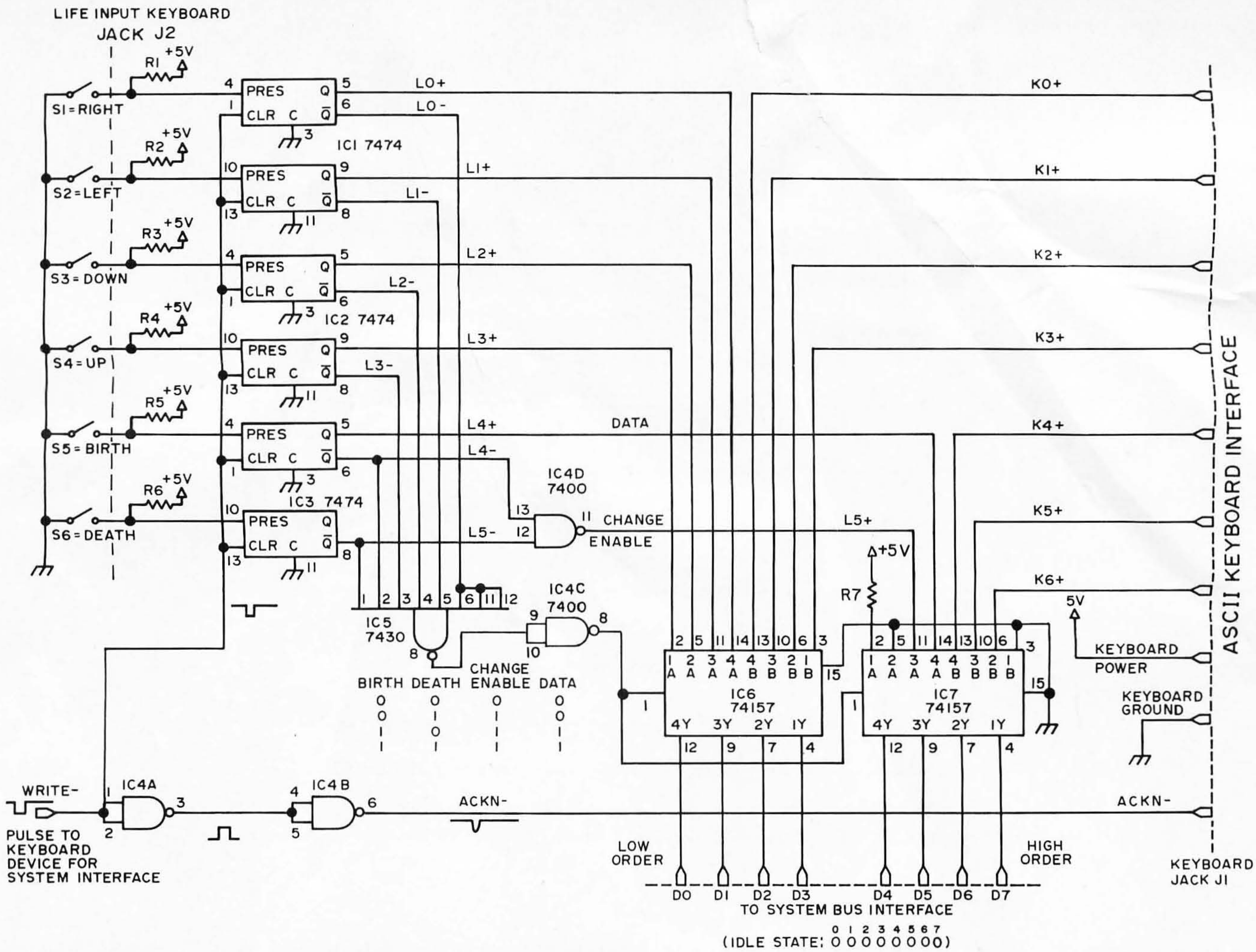


Figure 2: The logic diagram of a keyboard interface which implements the formats of figure 1. Resistors R1 to R7 are TTL pull up resistors. The value are not critical, and may range from 1 K0 to 10 K0, ¼ W.

otherwise the ASCII keyboard is connected and the cursor motion control keyboard is ignored. Note that the cursor motion input *has priority* over the ASCII keyboard since it controls the data selector.

Finally, to complete the interface logic sections A and B of IC4 are used to buffer the computer-generated keyboard WRITE- signal which occurs when the computer writes data into the keyboard location. This signal is used to reset the graphics control flip flops. The buffered version of the signal (pin 6 of IC4) is used to drive the acknowledge line of the ASCII keyboard unit. A separate buffer is recommended due to the unknown loading of the ASCII keyboard device. In LIFE Line's design of a program, the logic of the KEYBOARD_INTERPRETER procedure is used to manipulate the interface.

What is not shown in figure 2 is the actual system bus interface. The design of such an interface must be done consistent with a given computer's data bus. In the prototype system for LIFE Line, a Motorola 6800 computer's data bus, buffered by National DM8833 Tri State bus transceivers is used. The interface thus consists of two DM8833's used to drive the bus, plus the address selection logic needed to detect the address of the keyboard and produce the bus enable signal as well as the WRITE- signal. For a computer based upon a kit, the input port logic will be in a standard form designed by the kit manufacturer. What is needed is a parallel input port, which might already exist if your computer kit comes with a keyboard and parallel interface.

Notes on Assembly

The prototype version of the graphic control keyboard is illustrated in photo 1. The keys were made from conventional magnetic reed switches obtained from keyboard units found at a computer auction. Any single pole single throw keyswitch can be used; options on mounting are left to the ingenuity of the builder. The arrangement of keys shown in photo 1 is designed so that the cursor motion controls are at the top in

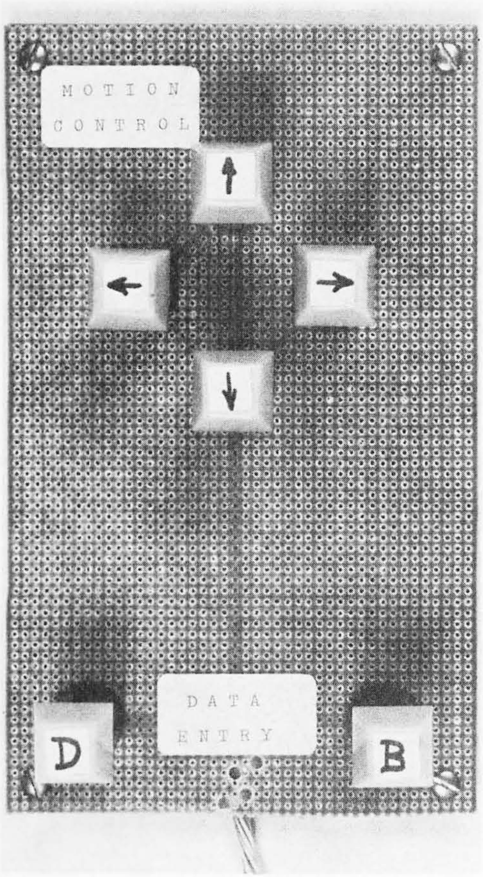


Photo 1: The graphics control keyboard of the LIFE Line prototype system. The group of four switches with arrows are cursor motion control keys. The two switches with captions "B" and "D" are the birth and death data keys, respectively.

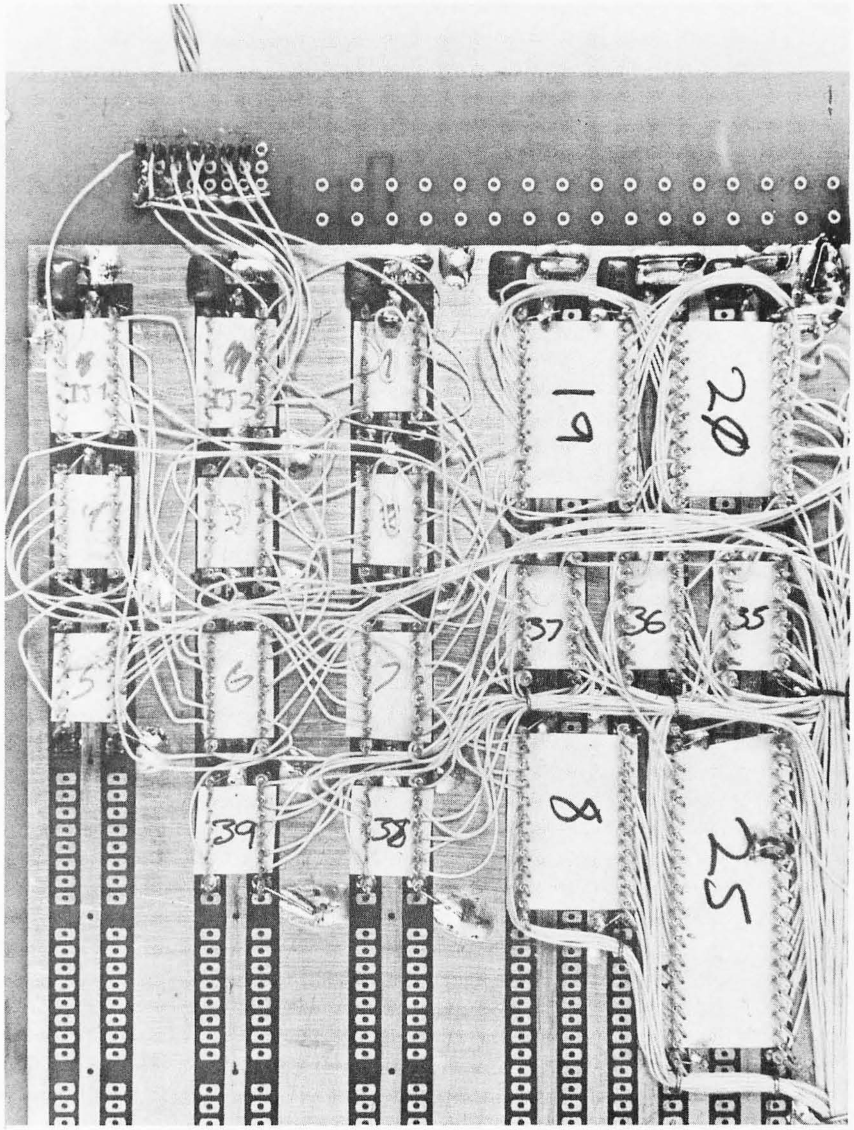


Photo 2: Detail illustrating wire wrapped assembly of figure 2 using a general purpose prototyping board.

a group of four. The arrows were applied using small pieces of self-sticking address labels of the type often used by computer centers. The two isolated switches at the bottom of this arrangement are the birth (B) and death (D) keys. The wiring of the keyswitches to the computer is accomplished through a multi-conductor bundle of wires trailing away at the bottom. This cable terminates in a dual-inline header plug which fits into a socket on the wire wrap board containing the computer and interface. Photo 2 illustrates the wire wrap wiring of the interface logic in the LIFE Line prototype system.

Using The Control Information

The purpose of the interface hardware is to combine two keyboards into a single input port with software distinguishing "who called" on the basis of the format shown in figure 1. How does the LIFE software handle this data format? Recalling the presentation in LIFE Line 3, the DEFAULT routine is called by the DECODE routine whenever DECODE cannot match an input from this port to an entry in the COMMAND table. Decoding of the graphics control format and ASCII numeric characters is left to the DEFAULT routine because of the systematic nature of these inputs.

How is this decoding done? One answer of course lies in the design of the DEFAULT routine. DEFAULT is specified in a procedure-oriented language in figure 3. Basically the DEFAULT processing follows one of two paths of execution according to the high order format identifier bit, bit seven of the formats illustrated in figure 1. The input data from the interface is passed to DEFAULT in the variable KEY which is set at line 11 of KEYBOARD_INTERPRETER (see LIFE Line 3, figure 3). The high order bit of KEY is tested by the AND operation of line 3. The masking bit string 10000000B

```
1   DEFAULT:
2     PROCEDURE;
3     IF (KEY AND 10000000B) NOT = 0 THEN
4       DO; /* GRAPHICS CONTROL INPUT CASE */
5         CALL MOVECURS; /* MOVE CURSOR PER INPUT */
6         IF (KEY AND 00100000B) NOT = 0 THEN
7           DO; /* CHANGE IS INDICATED */
8             IF (KEY AND 00010000B) = 0 THEN
9               CALL LPUT(XCOL, YROW,0); /* TURN OFF POINT */
10            ELSE
11              CALL LPUT(XCOL,YROW,1);/* TURN ON POINT */
12            CALL DISPLAY; /* SEND UPDATED LIFEBITS OUT */
13          END;
14      END;
15    ELSE
16      DO; /* ASCII NUMERIC DEFAULT CASE */
17        NUM = KEY – 30H;
18        IF NUM >9 THEN NUM = 9;
19        ZUM = 0;
20        DO FOR I = 1 TO 10; /* MULTIPLY = REPEATED ADD */
21          ZUM = ZUM + ENTRY;
22        END;
23        ENTRY = ZUM + NUM;
24        /* ENTRY NOW HAS NEXT DECIMAL DIGIT ADDED IN WITH */
25        /*     A BCD SHIFT BY ONE PLACE
26      END;
27    CLOSE DEFAULT;
```

Data (8-bit bytes) used by *DEFAULT* at this level:

NUM = temporary data byte used to hold a BCD digit for conversion to binary.

ZUM = temporary data byte used to form the product when ENTRY is shifted left 1 BCD digit by multiplication with 10, lines 20 to 22.

Data (8-bit bytes) used by *DEFAULT* but shared with the whole program.

KEY, XCOL, YROW, ENTRY

Subroutines Referenced by DEFAULT:

LPUT . . . Routine (used also by FACTS_OF_LIFE) which places the bit value of the third argument at a location specified by the first two arguments. Thus lines 9 and 11 define a new value for the bit at XCOL and YROW in the LIFEBITS matrix.

MOVECURS . . . The routine (see Fig. 11) which moves the cursor up, down, left or right depending upon the motion control switches which are read into the low order bits of KEY.

DISPLAY . . . The routine which copies LIFEBITS to the graphics output device for viewing.

Figure 3: The DEFAULT routine specified in a procedure-oriented language.

selects only the high order bit of KEY so that the result of the masked test will be zero if bit seven is zero, non zero if bit seven is one. If the result of the AND is not equal to zero, the graphics control case will be executed: the DO . . . END group extending from line 4 to line 14. If the result of the AND is zero, ASCII input is present so the character is forced into a numeric entry interpretation. The ELSE DO . . . END clause of lines 15 to 26 handles this alternative.

Graphics Control Processing

The processing of the graphics control format is not at all complicated. A procedure, called MOVECURS is executed first to decode the four low order bits of the graphics control format and adjust the cursor position.

MOVECURS is specified in a procedure-oriented language in figure 4. This routine contains four IF statements which test the four motion control bits. Motion is achieved for each logical one bit by simply adding or subtracting one from the corresponding cursor position variable XCOL or YROW. Note that this software takes care of an invalid combination of up and down (or left and right) in a unique way: nothing happens. If contradictory commands are input, the

ALTAIR 4K's

~~$264~~

$195!

The price of the ALTAIR 4K memory boards from MITS was incorrectly listed at $264 in the "MITS-MAS" Christmas Catalog printed in December's *Popular Electronics*. Unfortunately, the price change came after the catalog was printed.

The ALTAIR 4K board provides 4,096 words (bytes) of dynamic RAM (Random Access Memory) for the ALTAIR 8800. Each individual board contains memory protect circuitry and address selection circuitry for any one of 16 starting address locations in increments of 4K. The maximum access time of the ALTAIR 4K board was recently reduced from 300 nanoseconds to 200 nanoseconds.

Dynamic memory was chosen for the ALTAIR 4K board because it is generally faster and requires less current than static memory (a 4K static card that meets ALTAIR bus drive specifications requires close to three times the 5 volt current of dynamic memory).

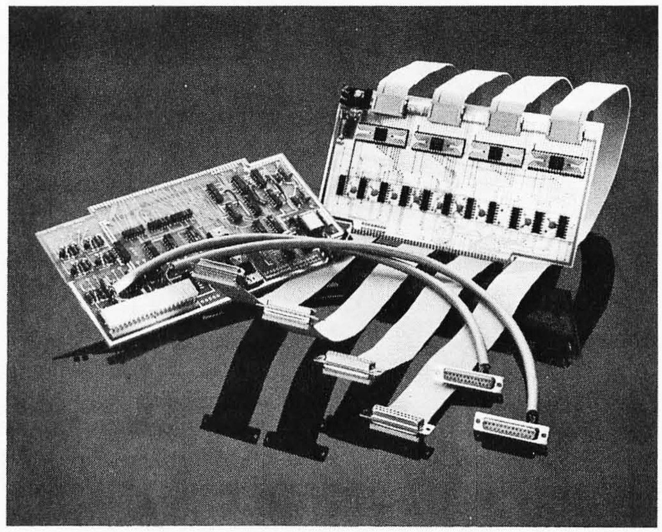
Other ALTAIR 8800 memory options include a 1K (1,024 words) and a 2K (2,048 words) static memory card. These low power static memory cards contain memory protect features and have a maximum access time of 850 nanoseconds.

ALTAIR 8800 Memory Prices:

4K Dynamic	**$195 kit and $275 assembled**
2K Static	**$145 kit and $195 assembled**
1K Static	**$ 97 kit and $139 assembled**

Prices, delivery and specifications subject to change.

NEW ALTAIR I/O CARDS



Two new ALTAIR 8800 I/O boards are now in production. These include a serial interface (88-2SIO) that can be ordered with one or two ports, and a parallel interface (88-PIO) that can be ordered with up to four ports.

Each port of the new Serial interface board is user-selectable for RS232, TTL, or 20 milliamp current loop (Teletype). The 88-2SIO with two ports can interface two serial I/O devices, each running at a different baud rate and each using a different electrical interconnect. For example, the 88-2SIO could be interfaced to an RS232 CRT terminal running at 9600 baud and a Teletype running at 110 baud.

An on-board, crystal-controlled clock allows each port to be set for one of 12 baud rates: 37.5, 75, 110, 150, 300, 450, 600, 1200, 1800, 2400, 4800, or 9600. Baud rates are the speed at which data is transfered between the computer and the I/O device.

Each port of the new Parallel interface provides 16 data lines and four controllable interrupt lines. Each of the data lines can be used as an input or output so that a single port can interface a terminal requiring 8 lines in and 8 lines out—or it can interface two input devices such as a paper tape reader and a keyboard—or two output devices such as a paper tape punch and printer—or any combination for custom applications. All data lines are fully TTL compatible.

ALTAIR 8800 Interface Prices:

88-2SIO Serial Interface	**$115 kit and $144 assembled**
88-SP additional 88-2SIO port	**$ 24 kit and $ 35 assembled**
88-4PIO Parallel Interface with one port	**$ 86 kit and $112 assembled**
88-PP additional 88-4PIO port	**$ 30 kit and $ 39 assembled**

MITS

"Creative Electronics"

MITS/6328 Linn N.E., Albuquerque, NM 87108 505/262-1951

```
MOVECURS:
    PROCEDURE;
    /* MOVE THE CURSOR BASED UPON THE FOUR LOW ORDER BITS */
    /* OF THE GRAPHICS CONTROL CHARACTER INPUT             */
    IF (KEY AND 1000B) NOT = 0 THEN
        YROW = YROW + 1;
    IF (KEY AND 0100B) NOT = 0 THEN
        YROW = YROW – 1;
    IF (KEY AND 0010B) NOT = 0 THEN
        XCOL = XCOL – 1;
    IF (KEY AND 0001B) NOT = 0 THEN
        XCOL = XCOL + 1;
    /* NOW, IF THE SELECTED KEY WAS ON, THE APPROPRIATE */
    /*  CURSOR POSITION REGISTER HAS BEEN CHANGED  */
    CLOSE MOVECURS;
```

Data (8-bit bytes) used by DEFAULT but shared with the whole program. See Table 2 of LIFE Line 3, BYTE #4, pg. 55, for details.

KEY, XCOL, YROW

Figure 4: The MOVECURS routine specified in a procedure-oriented language.

Decoding "who called" is done in software by the DEFAULT routine.

cursor position variable in question is both incremented and decremented with a net result of no change. Remember also that time delays are built into KEYBOARD_ INTERPRETER to govern the speed of changes when keys are held down continuously.

Upon return from MOVECURS with the newly updated position, the remaining portion of the graphics control processing consists of program logic to test for data entry. If the change enable bit (bit five) has a value of one, a change is indicated. Then if the data bit (bit four) is zero, the current position in LIFEBITS is turned off, indicating a death; if the data bit is one, the current position in LIFEBITS is turned on, indicating a birth. Graphics control change processing is completed at line 12 when DISPLAY is called to put the new data out on the display screen.

Numeric Default Processing

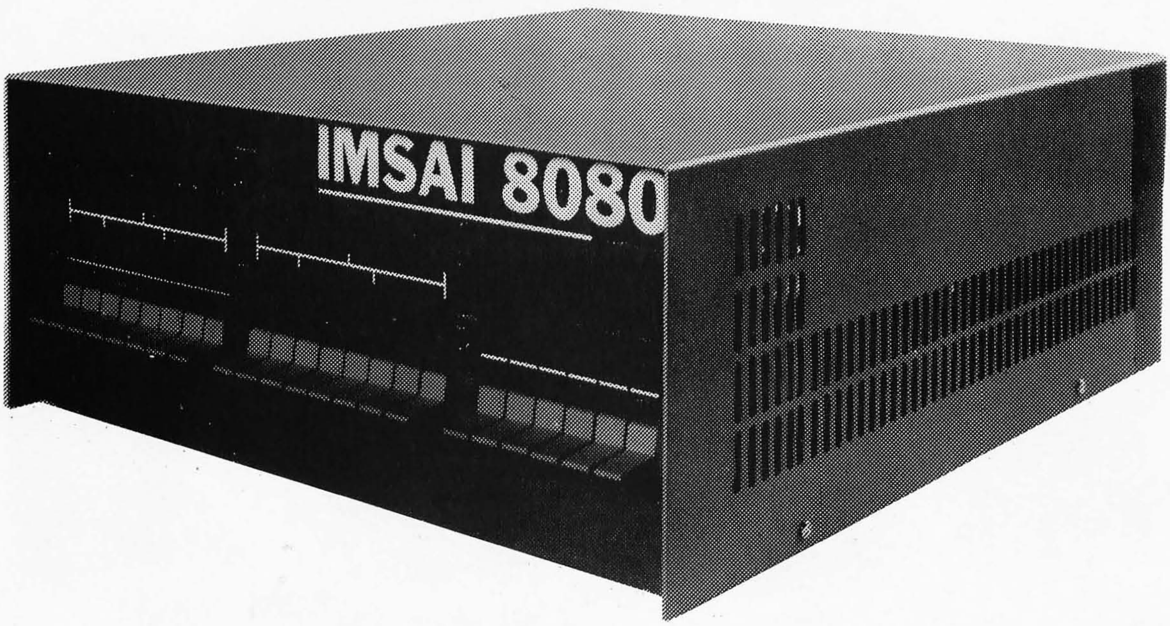
In the alternative DEFAULT processing case of an ASCII character which is not recognized by the DECODE routine, the program will *assume* numeric entry. In effect what this means is that any unrecognized non-numeric ASCII character will cause invalid data to be placed in the ENTRY register of the software since this little routine uses brute force to extract a numeric meaning. At line 17, a value of hexadecimal 30, denoted 30H, is subtracted from the key code. Since valid numeric ASCII characters run from hexadecimal 30 to 39, this will result in data running from 0 to 9 for valid numeric codes. The test of line 18 excludes invalid codes by forcing a 9 value. (Unsigned arithmetic is assumed here so that all 8-bit integer values not in the range 0 to 9 will be larger than 9.) Then the previous ENTRY value is multiplied by 10 using a repeated addition loop at lines 20 to 22. The new entry digit value is then added in to the low order at line 23. Note that ENTRY is a binary number, but that the digit being defaulted is entered with a decimal weighting. (For multiplication, an alternative to repeated addition in this case would be to observe that 10 x = 8 x + 2 x. Thus using three arithmetic left shifts both twice and eight times the original ENTRY could be obtained and summed producing the 10 x product.)

After execution of one or the other of the two paths determined by the format bit of the data in KEY, DEFAULT reaches its CLOSE statement and returns to DECODE.

Where Does LIFE Stand?

In the course of LIFE Line through this installment, the structure of the LIFE program has been the major topic. LIFE Line 4

IMSAI



A POWERFUL COMMERCIAL GRADE 8080 BASED MICROCOMPUTER SYSTEM REASONABLY PRICED

STANDARD FEATURES INCLUDE:

- A commercial grade cabinet, made of heavy gauge aluminum, custom designed for this specific product. Rack mounting is available as a special option.
- Lucite display panel with a photographically reproduced legend and LED display mask. In addition to standard lights and switches, the heavy duty front panel has an extra 8 program controlled LED's.
- Rugged commercial grade paddle switches backed up by reliable debouncing circuits.
- Gold plated contacts on board edge connectors to assure corrosion free contact at all locations.
- A heavy duty power supply to handle lots of extra cards, the power supply delivers up to 24 amperes at 8 volts, and 3 amps each at +16 and -16 volts.
- A sturdy card cage to properly support up to 22 printed circuit boards.
- Plug removable front panel board that eliminates the normal wiring harness between front panel and back plane. For special dedicated uses, the front panel board may be removed from the unit altogether and used only when program changes are desired.
- A full line of peripheral devices to build truly powerful and complete systems.
- An exclusive shared memory facility from the company that pioneered shared memory microprocessing. Shared memory facilities allow multiple microprocessors in the same cabinet to share the same memory, yet run different programs in parallel with each other.
- A full compliment of board options.

BASIC COMPUTER INCLUDES: 8080A Processor Board, Front Panel Control Board, lights and switches, power supply, Expander Board, Cabinet, . . . and . . . 256 BYTES OF RAM (on a 4K Board).

PRICES: KIT . . . $439 ASSEMBLED . . . $621

OPTIONS: Rack mount, RAM & PROM Boards, Parallel & Serial I/O Boards, an 8 level Priority Interrupt/Clock Board, Shared Memory Boards, Interface Boards.

PERIPHERALS: Floppy Disk Drives, 32 Column/125 LPM Printer, 80 Column/300 LPM Printer, 50 Megabyte Disk Drive, 30 character per second Printer with Plotting Mode.

TERMS: Check or Money Order, Bankamericard, Master Charge, 25% deposit on COD orders. On all orders under $1000, add 5% postage and handling. California residents add 6% tax.

SEND FOR FREE CATALOG OF IMSAI PRODUCTS

Contact: IMS ASSOCIATES, INC.
1922 REPUBLIC AVENUE
SAN LEANDRO, CA 94577
(415) 483-2093

SPECIAL NOTICE TO ALTAIR 8800 OWNERS:

If you would like to step-up to the commercial quality of an IMSAI 8080, you will be pleased to know that your ALTAIR 8800 boards are "plug-in" usable—without modification—in the IMSAI 8080 cabinet. Furthermore, by acquiring IMSAI's unique Memory Sharing Facility, your ALTAIR MPU board and IMSAI MPU board can co-exist in the same cabinet, operate in parallel with each other, and share all memory in common. This is the technology that laid the foundation for IMSAI's powerful HYPERCUBE Computer and Intelligent Disk systems (recently featured in Computerworld, Datamation and Electronics magazines).

has introduced the first hardware considerations — the special keyboard — as a requirement in the specification of graphics control processing. LIFE Line 5 will continue the software theme by completing the initial specification of the LIFE program design exclusive of the RESTORELIFE, SAVELIFE and INITIALIZATION procedures which together form a major software subject in their own right. LIFE Line 5 will cover the DISPLAY, RUN, SETXLOC, SETYLOC, LIFEDONE, and SETNGEN procedures as its main theme. Then the series will turn to the hardware of the LIFE system prototype in more detail, to provide a basis for the generation of actual executable programs which will run on the prototype system. The first major phase of the LIFE Line project will be completed when it is possible to draw a LIFE pattern on an oscilloscope output device using the cursor motion control keyboard, then initiate the pattern evolution according to the facts of LIFE as presented in LIFE Line 1.

The second major phase of the project will be the addition of the data management hardware and softwave facilities of the SAVELIFE, RESTORELIFE and INITIALIZE procedures. These facilities will enable the construction of initial patterns from "standard parts" saved on a mass storage device. As always, the aim of the entire series of LIFE Line articles is to show how the bits and pieces of hardware and software design fit together to produce a working application system. ■

A bibliography of Scientific American *information on LIFE (all references are to Martin Gardner's "Mathematical Games" column).*

October 1970: page 120. This is the original LIFE article, including the definition of the Facts of LIFE, and illustration of numerous fundamental patterns.

November 1970: page 118. Answers to several questions posed in the first article on the subject, including definition of the several varieties of "spaceships."

January 1971: pages 105, 106 and 108. Continued progress on the LIFE front including answers to several unsolved questions and results of a flurry of computer LIFE activity.

February 1971: Special "Mathematical Games" article on "cellular automata theory."

March 1971: pages 108 and 109. Short note about progress made by John Conway and R. William Gospers, plus illustration of a large scale flip flop pattern which is delicately balanced and easily destroyed by minor disturbances such as impact of a glider.

April 1971: pages 116 and 117. Examples of fuses, the five cell cross series, and announcement of Robert T. Wainright's LIFELINE newsletter.

November 1971: page 120. Short note on continued progress at the MIT AI Laboratory.

January 1972: page 107. The discovery of the "eater" by Bill Gospers at MIT.

This is an essential list of readily available information on the LIFE game which interested readers can research in any complete public or university library.

An Aside Regarding the Ultimate LIFE

LIFE on a 64 x 64 grid is an achievable project for the home brew computer enthusiast. But it is far from the ultimate. My thanks to Bob Clements of Lexington, Massachusetts for arranging a demonstration by R. William Gospers, Jr., at the MIT Artificial Intelligence Laboratory one recent Saturday evening. When LIFE was first widely publicized by Martin Gardner in his October 1970 Mathematical Games column in *Scientific American,* it helped set up a flurry of research work on the subject.

Bill Gospers and his associates at the MIT AI Lab took the definition of John Conway's game and began constructing a highly efficient LIFE system running on a Digital Equipment Corporation PDP-6 computer with a high resolution 1024 x 1024 position oscilloscope display. This research tool was used by the MIT people to generate numerous mathematically interesting LIFE patterns. These include such fundamental discoveries as glider guns, space ship factories, a binary transcendental number calculator, and a Turing machine pattern. The ultimate climax of the evening's demonstration was Bill's demonstration of a disproof — by example — of John Conway's conjecture that no LIFE pattern could grow without limit. The particular example he used is a colossal moving glider gun factory — a pattern which leaves a trail of active glider guns behind it as it travels slowly to the right on the display screen. This pattern fills the plane of the LIFE matrix with cells, and the number of active cells increases in proportion to the square of the number of generations the pattern has lived. After an arbitrary length of time, an arbitrary region of the plane will be filled with glider patterns emanating from the residue of glider guns produced by this LIFE machine.

The programs which form the MIT LIFE system are run on equipment far beyond the range of price a home brewer could consider — but with the advances in technology it is now possible to make a LIFE system which demonstrates many principles without breaking budgets.

Total Kitchen Information System

Ted M Lau
7740 P Chalmette Dr
Hazelwood MO 63042

I have become a hateful person just because my grocery list is unsorted.

I want to outline a plan for a total kitchen information system (TKIS) suitable for implementing on a home computer. This outline is the first step in the development of TKISs of arbitrary complexity from the simplest inventory modules to artificial intelligence modules (such as those suggested by Richard Gardner in the October 1975 issue of BYTE). The functional approach used here should allow the reader to plan a complex system using small and manageable, "byte-sized" pieces, or to interface independently developed modules.

This project began as a gripe list my wife and I compiled after many frustrating experiences in the kitchen; throwing out spoiled food we'd forgotten in the refrigerator, abandoning a recipe for lack of a key ingredient, reeling with confusion after reading pages of grocery specials, neither being able to remember an appealing recipe nor to find it among all our cookbooks, and so on.

Hierarchy Chart

Figure 1 shows the functions to be performed by a TKIS, structured in hierarchic fashion – meaning that every function is made up of several subfunctions, each function box performs one general task which can be divided into several specific tasks, and so on. This chart differs from a flowchart in that the function boxes are not necessarily performed in left to right order, nor are the conditions for execution given. The hierarchy (H) chart attempts to outline *what* a system must do, but not *how, when,* or *if.*

Each rectangle in the chart represents a transformation of some inputs into some outputs. For example, box 1.0 takes grocery prices from several markets and spits out a list of bargains to be scheduled into meals. Box 2.0 accepts a list of on-hand perishables, in addition to the output from 1.0, and yields a schedule of meals. Box 3.0 transforms the meal schedule into the food needed. Box 4.0 transforms raw, separate foodstuffs into cooked fare. Box 5.0 turns a meal into leftovers and garbage, and 6.0 turns garbage into cleanliness.

Notice that I've ignored inputs that appear unchanged as outputs, such as the recipes consulted to plan the meal (2.0): They are brought in at the beginning and returned unchanged at the end of the task. These unchanged or rarely changed inputs are the tables and files referenced by the function boxes. These tables and files appear to be internal to the boxes, and therefore can be ignored for the time being, thus allowing me to concentrate on TKIS functions. Though file design itself can be put off, provision must be made for the creation and maintenance of this data (7.0). Examples are the writing of recipes onto blank recipe cards, or the (presumed) structuring of a previously unstructured human brain nerve net to respond to a low price in hamburger.

Notice that action boxes (3.5, 4.5, 5.0) are mixed in with thought boxes. The H chart attempts to completely describe all the functions involved in operating a kitchen, whether primarily physical or primarily informational. While no one can seriously attempt to computerize these physical tasks at the present time, we must remember that all physical processes have informational

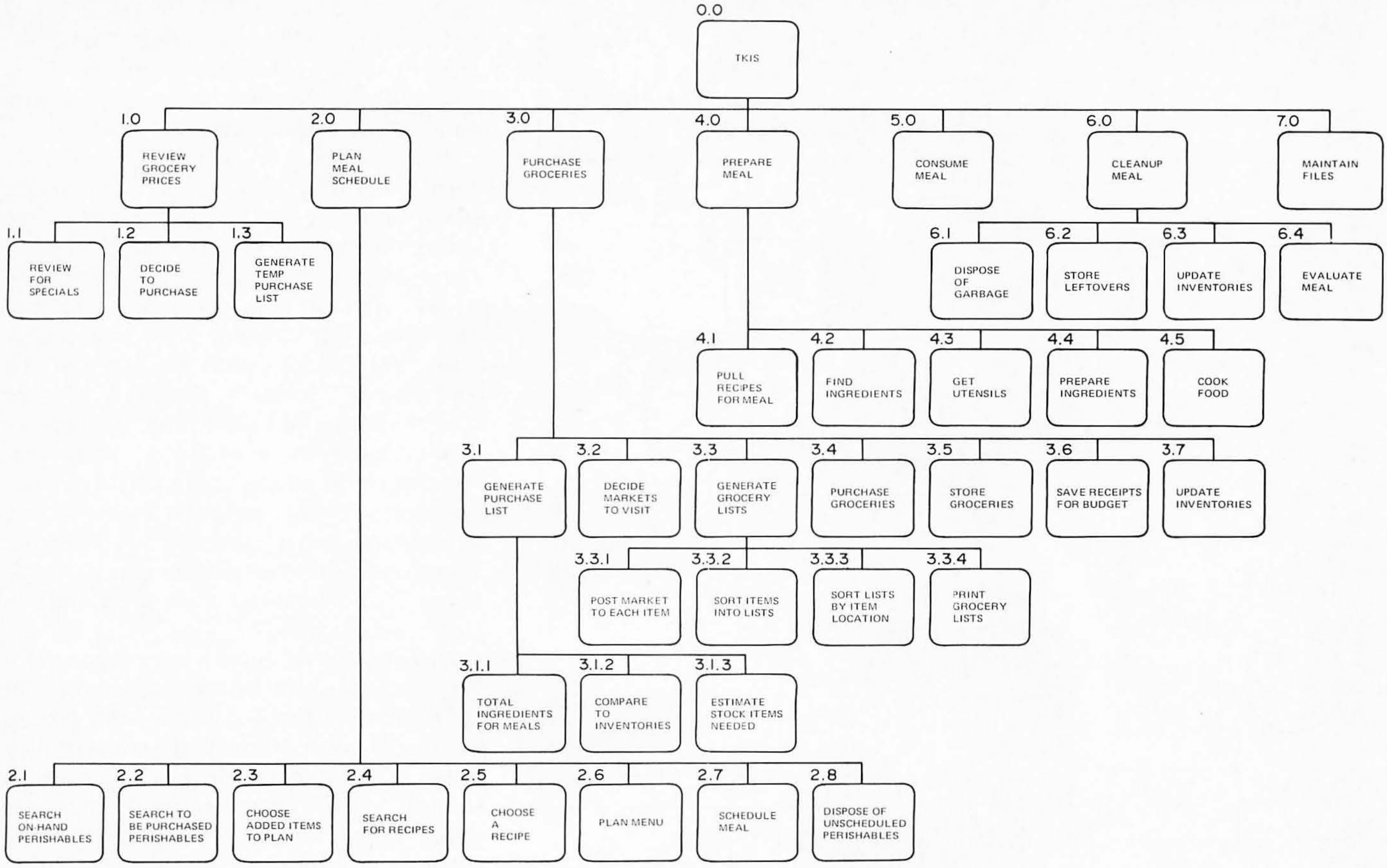


Figure 1: Functions of Total Kitchen Information System.

components (and are thus fair game for computer enhancement), and that any distinction between physical and informational is strictly provisional (and is subject to erosion as computers expand their capabilities to manipulate objects, as in robotics and automation). So the H chart incorporates into its comprehensive structure both modules that are subject to present data processing solutions and modules that must wait for future technology. (Readers will identify box 4.0 as the voice responsive vending machine in the rec room of the Star Ship Enterprise.)

Notice that the H chart says nothing about computers. It describes my conception of a very rigorous manual system that could be performed with paper and pencil. It purposely steers clear of computer concepts to allow you to be flexible in making software and hardware design decisions. To paraphrase: "Hardware and software may pass away, but functions endure." The tasks to be performed by the TKIS remain unchanged from one system configuration to another.

The H chart functionally describes my view of what must be done to get meals on the table. It is triggered by specials and perishables in that it tries to cut costs by planning meals using bargains, and to reduce wastage by scheduling perishables in timely fashion. Specifically, TKIS plans to review a large number of grocery item prices and to call attention to those that meet a *specials* criterion specified by the developer (1.1). It plans to call attention to items in inventory whose perishable date falls within the next meal scheduling period (2.1). It plans to retrieve recipes based on key ingredients and other characteristics such as casserole, quick-meal, Chinese, price-per-serving, nutritional values, etc., (2.4), and to reveal the recipe ingredients not on hand, or to reveal only those recipes whose ingredients are all on hand. It plans to help the kitchen operator decide which markets to visit by simulating the expenses of buying at various markets, including labor time and gasoline costs (3.2). It plans to calculate the quantities of ingredients needed for recipes with adjusted servings (3.1.1, 4.1). It plans to collect menu and recipe evaluations (yum, good or echch) (6.4), along with keeping past meal schedules and market receipts, for future analysis in planning menus, purchasing foods, budgeting, and dietetics. It even plans to sort the items on each grocery list into store location order, so that by walking through the store in a prescribed way the items will be encountered in order (3.3.3). This is big on my gripe list: I hate carrying a pencil to mark the groceries I buy, and I also hate chasing all over the store to find the last

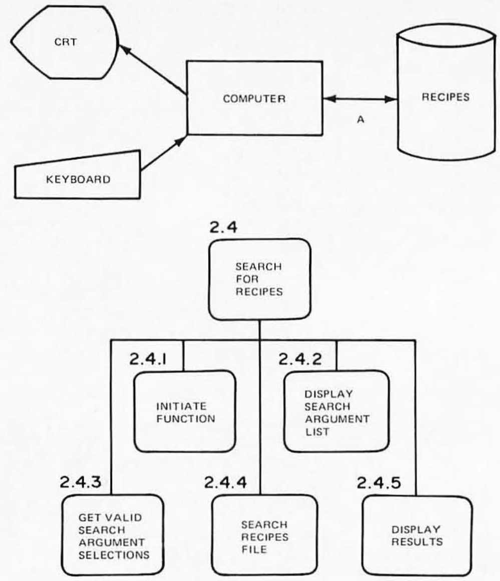


Figure 2: Recipe Subsystem.

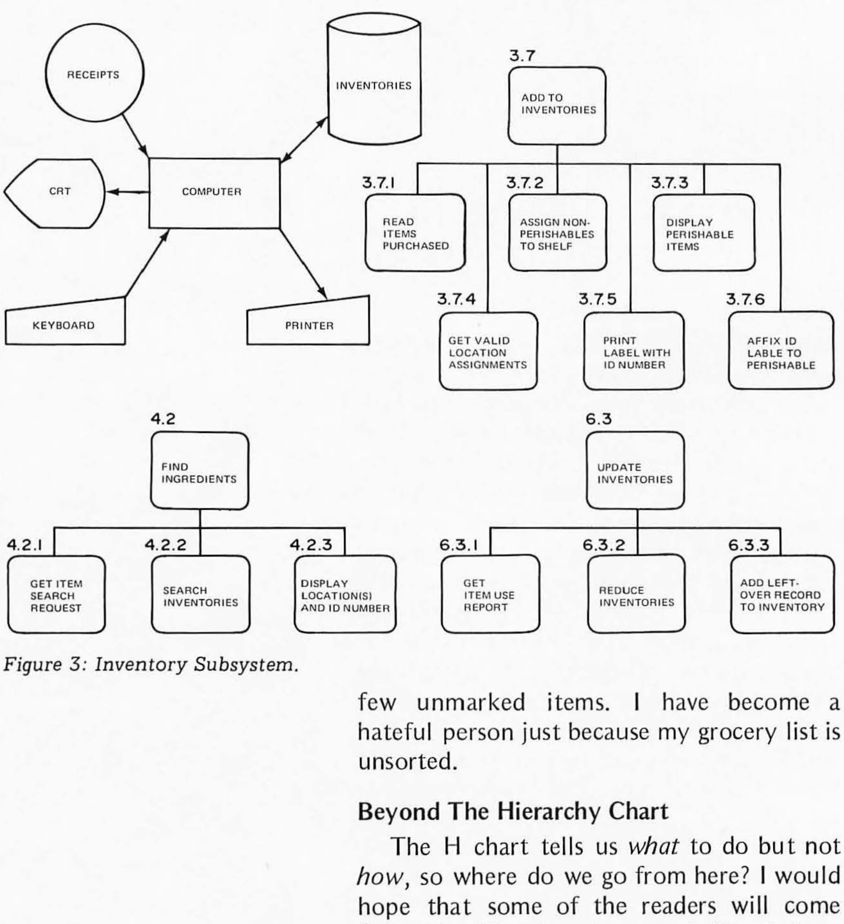


Figure 3: Inventory Subsystem.

few unmarked items. I have become a hateful person just because my grocery list is unsorted.

Beyond The Hierarchy Chart

The H chart tells us *what* to do but not *how*, so where do we go from here? I would hope that some of the readers will come forward with data base and file designs, hardware specifications, and program descriptions. This is a massive project and certainly in need of special talents and diverse opinions. There are many well known techniques for designing computer systems, and I think it is sufficient for me to mention some of the potential problems that may be encountered.

1. Is the proposed TKIS technically and economically feasible in a home? If not now, will it become so in a time frame approximately equal to the development time? To answer these questions, someone must expand the effort to prepare detailed hardware, software, and manpower estimates.

For example, a recipe retrieval subsystem might use a CRT with keyboard to initiate a search of the recipe file, and to display the results on the cathode ray tube (CRT) (see figure 2 for a schematic of the hardware and an H chart of the software functions). Assuming a file contains 2000 recipes averaging 500 B each, what are the cost and performance tradeoffs between a tape system versus disk system? In order to make this estimate we must know, first, what response time is acceptable to the kitchen operator. If the program reads records sequentially, what tape speed is required? What bandwidth is needed for data path A in figure 2 and how fast must programs execute? And so on . . .

2. How can the human labor required for data entry be kept below that required for the manual system? Data entry is the process by which humans, through the sweat of their brows, convert data into machine readable form so that the computer can do marvelous things with it and look like a genius. Data entry is probably one of the most costly items in the operating budget of the TKIS, and certainly one of the most boring.

Table 1 lists the tables and files needed to implement a basic version of the TKIS. Of the tables, *recipes* and *prices* represent large data entry tasks at initial system startup and at periodic intervals. It would be very nice if the kitchen operator could acquire data in machine readable form (on cassettes or via the phone line). The book and magazine publishers could supply a periodic update of recipes and the markets an update of prices. Standard formats would have to be developed for these interfaces, and a customer base must be developed to provide an economic incentive.

The files, on the other hand, originate within TKIS and change continually with use, making it difficult to solve the data entry problem in the same way. For example, figure 3 describes a test design for an inventory module. The functions of this subsystem are to add a record (or a count) of each purchased item to the inventory file corresponding to the storage location, to allow retrievals by item, and to decrement the inventories as items are used. The major data entry requirements are to tell the computer what was used and what was bought.

An efficient way to do the former is to *signify* what was used, instead of *specifying* in detail what was used. By entering the recipe name (say, recipe B), the operator

Table 1: Functional Storage Requirements.

File Name	Contents	Possible Source
A. Tables		
1. Prices	grocery item prices by brand for each market	Grocers
2. Recipes	ingredients, instructions, recipe characteristics, nutritional data, number of servings	Book and Magazine Publishers
3. Menus	groups of recipes, menu characteristics	Book and Magazine Publishers
4. Calendar	dates, meal times, number of guests, other requirements	TKIS User
5. Markets	market name, address, distance	TKIS User
B. Files		
1. Inventories	number and quantity of ingredients by location	TKIS User
freezer } refrigerator }	also perishable items by ID No.	
shelf		
stock items –	number and quantity of items, rate of use (salt, soy sauce)	
2. Meal Schedule	menu or recipes for each meal	TKIS User
3. History	past schedules and evaluations, market receipts, etc.	TKIS User
4. Working Storage	purchase list, grocery lists, etc.	TKIS User

says in effect that "the ingredients for recipe B were used." This requires that the computer have a recipe file for translating "recipe B = ingredients D, E, F." If the computer lacks this file, the operator must enter the specific ingredients used. Thus a stand alone inventory subsystem is less data entry efficient than one integrated into a full TKIS (a truism about systems in general).

On the other hand, telling the computer what was bought can be handled rather neatly, by adhering to the rule that once the data is in machine readable form it should not be degraded out of same. Instead of a paper receipt, the bag person at the market will plop a cassette in your bag containing all the items you purchased and their prices. This cassette will have been produced by the market's point-of-sale terminal which so graciously performed the data entry chore for you by optically scanning your groceries. (In fact – or rather in fantasy – the market won't even have to provide the cassette: you will bring the purchase list created by TKIS on cassette to the store, insert it into the computer at the front door which sorts and prints your grocery list in location order (3.3.3 and 3.3.4), and carry the cassette to the checkout counter for recording of your receipt.)

3. What does the kitchen operator do when the system goes down because a disk *crashes*, or the bus turns *flaky*, or a program *blows up*? (This picturesque lingo seems to less accurately describe the condition of the computer than it does our emotional state after the unthinkable has happened.) Backup manual procedures or alternate computer services must be provided to allow the kitchen operator who has become dependent on the TKIS to function while the system is down. Adequate system recovery and restart procedures must be designed, and a technique developed for catching the computer up on what transpired while it was unconscious. The importance of these considerations will depend upon the complexity and reliability of the hardware and software, but must be conceived and designed as an integral part of the total system.

4. Finally, assuming a TKIS was developed, would a kitchen operator use it? Besides being more efficient, less costly, and all the other good reasons for which we developed it, the TKIS must be flexible enough to allow for human inefficiency and taste preferences. What if the TKIS user doesn't want to prepare the scheduled meal for the evening? TKIS must be able to take account of human inconsistency.

Summary

I have briefly outlined the functions I think a kitchen information system should perform, and mentioned some considerations affecting its design. I hope this article will help catalyze development efforts in what appears to be a fruitful home computer applications area. ■

Golf Handicapping

Dr. George Haller
1500 Galleon Dr.
Naples FL 33940

Almost every golf club keeps a roster of its members with up-to-date golf handicaps. By using these handicaps, the members with varying degrees of skill in the game can play competitively against each other. Many large clubs have gone to semiautomatic computer services with terminals in the clubhouse but most clubs still appoint a handicap committee of members which meets periodically, usually monthly, to calculate each member's handicap.

Basically, a golf handicap is calculated by taking the average of the best ten of the twenty most recent scores for a regulation 18 hole game, subtracting the course rating from this average, and multiplying this difference by a factor of 0.85. This result is rounded off to the nearest whole number to become the handicap. As an example, if the average of the player's best ten of his most recent scores is 95.2 and the course rating is 68.6, then the difference of 26.6 is multiplied by 0.85 giving 22.6. Rounding gives this player a handicap of 23.

The calculating of these handicaps offers a small business opportunity to the owner of a small computer. Sheets can be made up with spaces for the members' names and columns for the most recent twenty scores, the old and new handicaps. The handicap

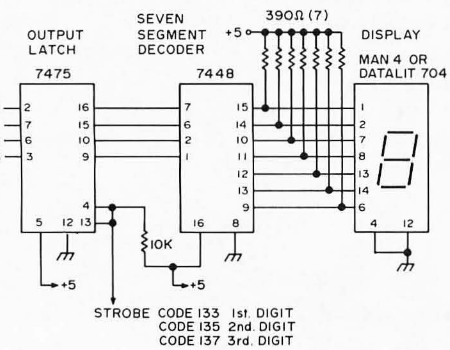


Figure 1: Each one of the three digits of the special display for this program is wired according to this schematic. One output port is required for each digit.

DUFFER Program Symbol Table: In order of appearance,

DUFFER	002/000	ONCHO	002/217
NEXTIN	002/005	SUBRATE	002/221
INTEN	002/036	NOBRRW	002/230
INUNIT	002/056	INCRD	002/327
GOAHEAD	002/071	ADD200	002/331
DROUTN	002/114	GODIVI	002/334
RSTRT	002/126	DIV1	002/341
CALCULA	002/134	ENDIVI	002/363
BUBBLE1	002/141	SCORE	003/000
BUBBLE2	002/146	INCRC	003/040
SWITCH	002/164	RNDLOOP	003/042
ADDSCOR	002/175	FRSTRND	003/050
ADDLOOP	002/203	ENDRNDT	003/062

Table 1: The absolute and symbolic assembly listing of the DUFFER program written for an 8008 computer.

```
address   octal code   label     op.  operand      commentary

002/000   056 003      DUFFER    LHI  H(SCORE)     set up address of the
002/002   066 000                LLI  L(SCORE)        score data;
002/004   125                    OUT  12           SCELBI keyboard handshake;
002/005   111          NEXTIN    IN   2            read SCELBI keyboard;
002/006   240                    NDA               test status in bit 7;
002/007   120 005 002            JFS  NEXTIN       loop til ready;
002/012   125                    OUT  12           SCELBI keyboard handshake;
002/013   074 320                CPI  "P"          is it a "P"?
002/015   150 134 002            JTZ  CALCULA      if so then go process scores;
002/020   016 000                LBI  0            clear input register B;
002/022   074 261                CPI  "1"          is it a "1" digit?
002/024   110 044 002            JFZ  OUTTEN       if not then skip next digit;
002/027   133                    OUT  15           send to special display;
002/030   016 144                LBI  100D         initialize 100 decimal;
002/032   300                    NOP
002/033   300                    NOP               several NOPs here and there
002/034   300                    NOP                  give  room for mistakes;
002/035   300                    NOP
002/036   111          INTEN     IN   2            read SCELBI keyboard again;
002/037   240                    NDA               test status in bit 7;
002/040   120 036 002            JFS  INTEN        loop for tens digit;
002/043   125                    OUT  12           SCELBI keyboard handshake;
002/044   135                    OUT  16           tens to special display;
002/045   044 017                NDI  00001111B    mask BCD digit  in low order;
002/047   022                    RAL               2 times tens digit;
002/050   320                    LCA               save in C register;
002/051   022                    RAL               2 x 2 x tens digit;
002/052   022                    RAL               2 x 2 x 2 x tens digit = 8 x digit;
002/053   202                    ADC               (2 + 8) x tens digit = 10 x digit;
002/054   201                    ADB               add in 100s or zero from B;
002/055   310                    LBA               save 10s + 100s in B;
002/056   111          INUNIT    IN   2            read SCELBI keyboard again;
002/057   240                    NDA               test status bit 7;
002/060   120 056 002            JFS  INUNIT       loop for units digit;
002/063   125                    OUT  12           SCELBI keyboard handshake;
002/064   137                    OUT  17           units to special display;
002/065   044 017                NDI  00001111B    mask BCD digit in low order;
002/067   201                    ADB               100s + 10s + 1s is the number;
002/070   310                    LBA               save binary score in B again;
002/071   111          GOAHEAD   IN   2            look for command code;
002/072   240                    NDA               test status as usual;
002/073   120 071 002            JFS  GOAHEAD      loop for D or C command;
002/076   125                    OUT  12           SCELBI keyboard handshake;
002/077   074 304                CPI  "D"          is it a "D"?
002/101   150 114 002            JTZ  DROUTN       if so then go deposit score;
002/104   074 303                CPI  "C"          else is it a "C"?
002/106   150 126 002            JTZ  RSTRT        if so then go ignore entry;
002/111   110 071 002            JFZ  GOAHEAD      loop til proper D or C is found;
002/114   371          DROUTN    LMB               deposit score in memory location;
002/115   060                    INL               increment L register;
002/116   006 013                LAI  "]"          define "]" pattern;
002/120   133                    OUT  15           "]"  into 100s digit;
002/121   135                    OUT  16           "]"  into 10s digit;
002/122   137                    OUT  17           "]"  into 1s digit;
002/123   104 005 002            JMP  NEXTIN       begin input sequence again;
002/126   006 012      RSTRT     LAI  "["          define "[" pattern;
002/130   133                    OUT  15           "[" into 100s digit;
002/131   104 005 002            JMP  NEXTIN       begin input sequence again;
002/134   006 014      CALCULA   LAI  "⊔"          define "⊔" pattern;
002/136   133                    OUT  15           "⊔" into 100s digit as marker;
002/137   061                    DCL               start the sort of scores;
002/140   336                    LDL               counter gets initial pointer;
002/141   056 003      BUBBLE1   LHI  H(SCORE)     pointer to beginning of SCORE;
002/143   066 000                LLI  L(SCORE)     also part of SCORE address;
002/145   323                    LCD               current pointer to C;
002/146   307          BUBBLE2   LAM               fetch current byte;
002/147   240                    NDA               set conditions;
002/150   060                    INL               point to next byte in bubble;
002/151   277                    CPM               compare next byte in bubble;
002/152   140 164 002            JTC  SWITCH       switch if need be;
002/155   021                    DCC               decrement sort count;
002/156   110 146 002            JFZ  BUBBLE2      keep going on inner loop of sort;
002/161   104 175 002            JMP  ADDSCOR      leave sort when done;
002/164   317          SWITCH    LBM               bubble sort involves a process
002/165   061                    DCL                  of switching neighboring
002/166   307                    LAM                  out of sequence pairs
002/167   371                    LMB                  until a uniform sequence
002/170   060                    INL                  is obtained; this set of
002/171   370                    LMA                  instructions does the switch;
002/172   104 141 002            JMP  BUBBLE1      then keeps trying until sort done;
002/175   300          ADDSCOR   NOP               spare byte with NOP;
002/176   016 000                LBI  0            B will accumulate carries;
```

```
002/200  026 011                LCI  9D         ten = nine plus predefinition;
002/202  307                    LAM             fetch first value before loop;
002/203  061         ADDLOOP    DCL             point to next value;
002/204  207                    ADM             add it from memory;
002/205  142 217 002            CTC  INCHO      if carry increment high order;
002/210  021                    DCC             decrement loop count;
002/211  110 203 002            JFZ  ADDLOOP    keep adding until done;
002/214  104 221 002            JMP  SUBRATE    branch around subroutine INCHO;
002/217  010         INCHO      INB             increment B subroutine
002/220  007                    RET                is simplicity squared;
002/221  240         SUBRATE    NDA             set flags;
002/222  024 267                SUI  267        octal low order of course rating;*
002/224  100 230 002            JFC  NOBRRW     if no borrow then no adjustment;
002/227  011                    DCB             high order adjust if borrow;
002/230  320         NOBRRW     LCA             save low order in C;
002/231  301                    LAB             high order prepared;
002/232  024 002                SUI 2           octal high order course rating;*
002/234  310                    LBA             save high order in B;
002/235  342                    LEC             start multiply by 8.5;
002/236  002                    RLC             shift left high order of data;
002/237  330                    LDA             save result in D;
002/240  320                    LAC             fetch the low order of data;
002/241  240                    NDA             clears condition flags;
002/242  002                    RLC             shift left low order;
002/243  142 327 002            CTC  INCRD      if carry then increment high order;
002/246  320                    LCA             D/C now have twice the result;
002/247  303                    LAD             fetch high order;
002/250  002                    RLC             multiply by 2 (4 x total);
002/251  330                    LDA             save high order;
002/252  302                    LAC             fetch low order;
002/253  240                    NDA             clear flags;
002/254  022                    RAL             multiply by 2 (4 x total);
002/255  142 327 002            CTC  INCRD      if carry then increment high order;
002/260  320                    LCA             save low order again;
002/261  303                    LAD             and again fetch high order;
002/262  002                    RLC             and multiply by 2 (8 x total);
002/263  330                    LDA             and again save high order;
002/264  302                    LAC             and again fetch low order
002/265  002                    RLC             and again multiply by 2 (8 x total);
002/266  142 327 002            CTC  INCRD      if carry then increment high order;
002/271  044 370                NDI  370        mask low order in 8 x  total;
002/273  320                    LCA             save low order in C again;
002/274  304                    LAE             fetch old original low order;
002/275  012                    RRC             divide by 2 giving .5 x;
002/276  044 177                NDI 01111111B   turn off high order bit;
002/300  340                    LEA             save in E
002/301  301                    LAB             fetch original high order result;
002/302  240                    NDA             clear flags;
002/303  012                    RRC             divide by 2 giving .5 x;
002/304  310                    LBA             save high order in B
002/305  304                    LAE             fetch low order;
002/306  142 331 002            CTC  ADD200     if carry, adjust low order;
002/311  240                    NDA             clear flags;
002/312  202                    ADC             8 + 0.5 = 8.5 x result;
002/313  320                    LCA             save low order 8.5x in C;
002/314  142 327 002            CTC  INCRD      increment D if carry to high order;
002/317  301                    LAB             fetch high order;
002/320  044 177                NDI  01111111B  clear bit 7;
002/322  203                    ADD             add other high order giving 8.5 x;
002/323  310                    LBA             save back into B;
002/324  104 334 002            JMP  GODIVI     branch around two subroutines;
002/327  030         INCRD      IND             subroutine increments D register;
002/330  007                    RET             and returns almost instantly;
002/331  004 200     ADD200     ADI  10000000B  turn on high order bit if off;
002/333  007                    RET             and also return almost instantly;
002/334  036 144     GODIVI     LDI  100D       load 100 count in D;
002/336  046 011                LEI  9D         load 9 count in E;
002/340  301                    LAB             fetch high order of numerator;
002/341  310         DIV1       LBA             save high order in B;
002/342  302                    LAC             fetch low order to A;
002/343  022                    RAL             divide by 2;
002/344  320                    LCA             save low order;
002/345  041                    DCE             decrement subtraction count;
002/346  150 363 002            JTZ  ENDIVI     if zero, then end divide;
002/351  301                    LAB             fetch old high order;
002/352  022                    RAL             rotate left;
002/353  223                    SUD             here is the repeated subtract;
002/354  100 341 002            JFC  DIV1       if no borrow then trial ok;
002/357  203                    ADD             else fix up for false subtract;
002/360  104 341 002            JMP  DIV1       and go back for more subtracts;
002/363  022         ENDIVI     RAL             multiply low order by 2;
002/364  340                    LEA             save low order in E;
002/365  006 377                LAI  11111111B  load all ones into A;
002/367  252                    XRC             exclusive OR with all ones
002/370  320                    LCA                complements C;
002/371  104 050 003            JMP  FRSTRND    jump around data;

003/000              SCORE      BLK  32D        32 bytes for scores;**

003/040  020         INCRC      INC             subroutine to increment C
003/041  007                    RET                and then return;
003/042  024 012     RNDLOOP    SUI  10D        subtract 10 from negative of
003/044  030                    IND                low order, increment couht;
003/045  104 062 003            JMP  ENDRNDT    go to end test;
003/050  301         FRSTRND    LAB             fetch high order;
003/051  240                    NDA             clear flags;
003/052  074 062                CPI  50D        compare to decimal 50;
003/054  102 040 003            CFC  INCRC      if =>50 then increment quotient
003/057  302                    LAC             low order to A;
003/060  036 000                LDI  0          clear divide count in D;
003/062  240         ENDRNDT    NDA             clear flags;
003/063  074 012                CPI  10D        is it less than 10?;
003/065  100 042 003            JFC  RNDLOOP    if not back for adjustment;
003/070  137                    OUT  17         send A to units digit output;
003/071  303                    LAD             fetch tens digit result;
003/072  135                    OUT  16         send D to tens digit output;
003/073  104 000 002            JMP  DUFFER     go back to consider next duffer;
```

*NOTE: The course rating in this example is assumed to be 69.5. To account for 10 double precision adds, this is multiplied by 10 giving 695. In octal, this takes two bytes with 002 in the high order, 267 in the low order.

** NOTE: The mnemonic "BLK" is used to indicate that a block of memory is reserved for use by the program, with the name (SCORE here) used to reference the first address in the block.

service furnishes these sheets to the golf club to fill in all the information except the new handicap. The handicap service then collects the sheets on a monthly or other periodic basis and returns them promptly with the newly calculated handicap. If done on a monthly basis, a reasonable charge might be from one to three dollars per member per year depending on the length of the season. This income will not support much of a computer service establishment, but could help in the purchase of some new peripherals.

In the author's case a SCELBI computer with its 8008 CPU was used with a standard ASCII keyboard and a small home brew readout which cost less than ten dollars for parts. A program equivalent to this one can be written for almost any small home computer. The diagram of the readout for each digit is shown in figure 1.

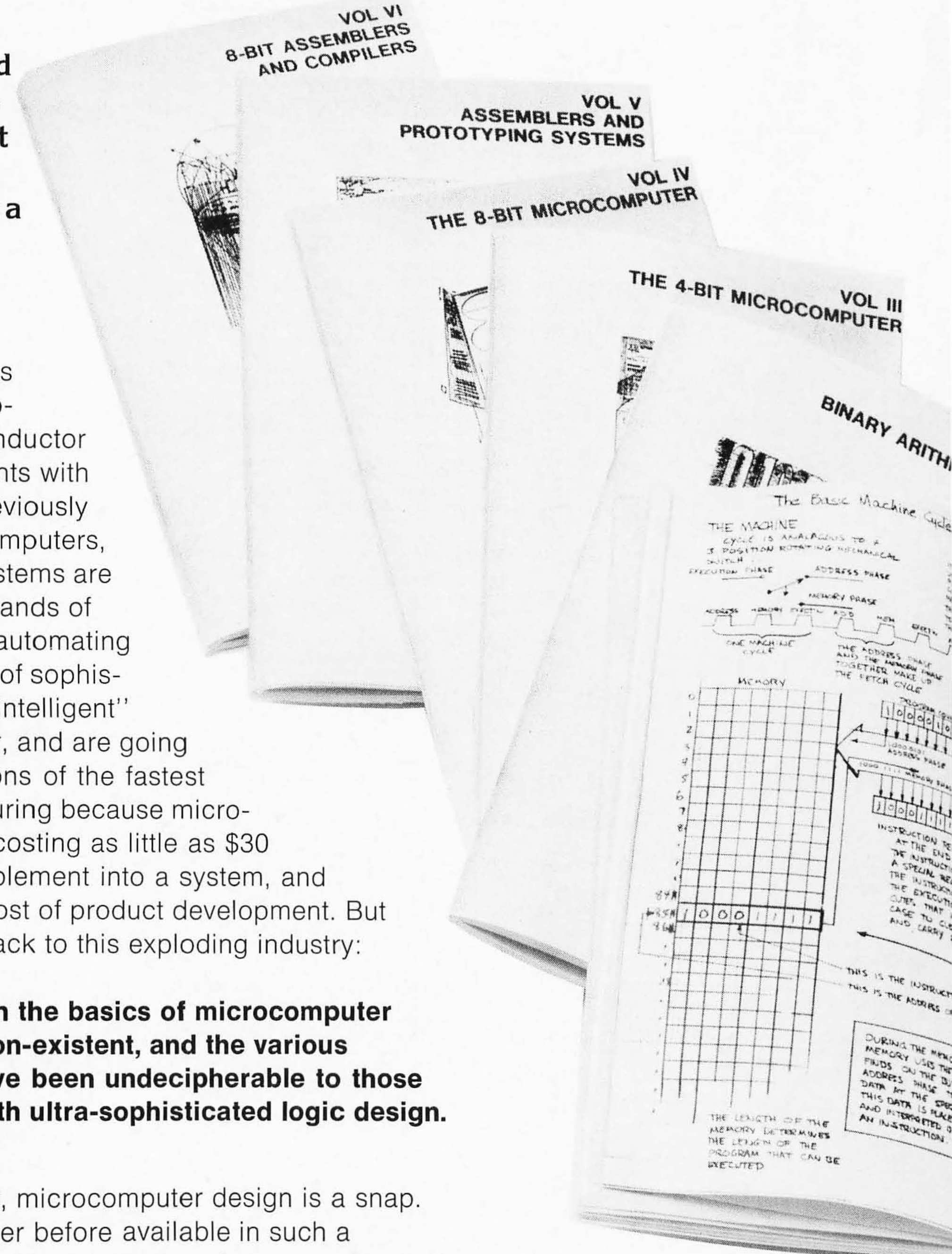
The program requires fewer than 320 bytes. Golf scores can range from 60 for a super player to 150 for a complete duffer. Regulation golf courses have ratings ranging from 65 to 75 which are always a matter of record at the golf club.

The program to calculate the handicap on an 8008 is shown in table 1. The keyboard is used to input the scores one at a time. If the first digit is a 1 then 100 (decimal which is equal to 144 octal) is loaded into register B and displayed on the left readout. If the first digit is not a 1 then the score is less than 100, and this digit is considered a second digit. The second digit is displayed on the central readout. Multiplied by 10, it is then added to the contents of register B. The final digit is shown in the right readout and added to the contents of register B. The operator verifies the score shown on the three digit readout. If it is correct, the key D is pressed which stores the score in memory and sends the program back to look for the next score to be entered. If an error has been made, by pressing key C the program goes back for a corrected score without storing the incorrect score in memory. After twenty scores are entered, the key P starts the calculating routine to process the scores. This routine sorts the scores in descending order, adds the ten lowest scores, subtracts ten times the course rating, multiplies by 8.5, divides by 100, rounds off the result and displays the handicap.

On Jan. 1, 1976, the multiplying factor changes from 0.85 to 0.96. To adapt the program, take the difference as obtained at location 002/233, multiply by 4, add 50, divide by 100, subtract the quotient (disregard the fraction) from the original difference, add 5 and divide by 10. The quotient, dropping the fraction, will be the rounded off handicap. ■

Microcomputer

A new six-volume programmed learning course from Iasis tells you EVERYTHING about what microcomputers are and how you can design and implement a microprocessor-based system.



Since the transistor was invented, no single electronics innovation has made such an impact as the microcomputer. Powered by tiny semiconductor chips containing computing elements with the same powers and functions previously found only in large scale digital computers, these dedicated microcomputer systems are now being applied to literally thousands of applications. Microcomputers are automating assembly lines, providing the heart of sophisticated electronic games, making "intelligent" computer peripherals even smarter, and are going so far as streamlining the operations of the fastest food chains. This revolution is occuring because microcomputers are very inexpensive—costing as little as $30 in production volume—easy to implement into a system, and significantly reduce the time and cost of product development. But there has been one serious drawback to this exploding industry:

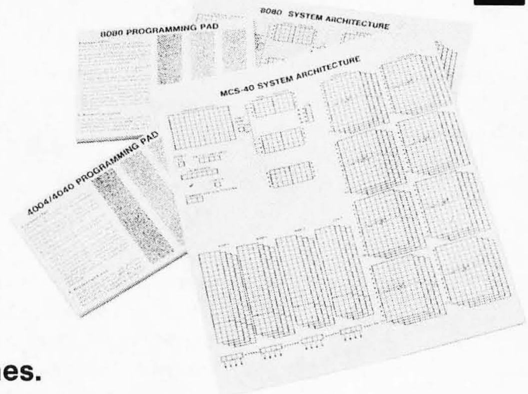
Training materials and courses in the basics of microcomputer technology have been virtually non-existent, and the various published manuals and texts have been undecipherable to those not already intimately familiar with ultra-sophisticated logic design.

Once a designer has the hang of it, microcomputer design is a snap. But without the fundamentals—never before available in such a readable, understandable and simplified format—microcomputer design has been unbelievably difficult. The comprehensive, step-by-step six-volume Programmed Learning Course on Microcomputers from Iasis makes the unbelievably difficult almost ridiculously simple. The authors of these texts have been involved on a professional level in the microcomputer industry since it became an industry. Their direct, first-hand experience in the whys, hows, wherefores and potentials of microcomputers

Iasis Texts:
1. Binary Arithmetic
2. Microcomputer Architecture
3. The 4–Bit Microcomputer
4. The 8–Bit Microcomputer
5. Assemblers and Prototyping Systems
6. 8–Bit Assemblers and Compilers

Finally, you can get a comprehensive training course on microcomputers that puts all the hard-to-get information at your fingertips in an easy-to-read, easy-to-understand and even easier-to-implement manner. You can get it here...now.

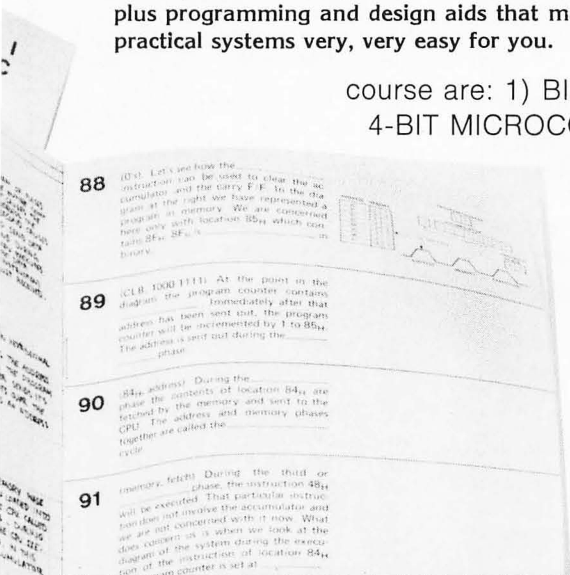
design is a snap.



have made this six-volume collection the most valuable and meaningful series ever published on microcomputer design. The books combine the most effective methods of programmed instruction with the entire gamut of essential information vital to the designer of a micro-based system. You begin with the ABC's of microcomputers and go through a virtual post-doctoral course... **and the unique, self-testing programmed learning Iasis course enables you to understand and absorb every bit of the information every step of the way through the six volumes.**

The Iasis course gives you more than 700 pages of detailed, illustrated microcomputer information—including more than 1,700 self-tests you use to evaluate your progress—plus programming and design aids that make the design of practical systems very, very easy for you.

Specific details are provided on four of the industry's most versatile microcomputers—the 4004, 4040, 8008 and 8080 from Intel Corporation—but the basic design information will apply to any and all microprocessors. The six volumes you receive with the course are: 1) BINARY ARITHMETIC; 2) MICROCOMPUTER ARCHITECTURE; 3) THE 4-BIT MICROCOMPUTER; 4) THE 8-BIT MICROCOMPUTER; 5) ASSEMBLERS AND PROTOTYPING SYSTEMS; and 6) 8-BIT ASSEMBLERS AND COMPILERS. Plus, this detailed course provides you with two programming pads and two simplified design aids so you may quickly and easily develop both 4-bit and 8-bit microcomputer systems. Use the coupon below to order your course from Iasis, Inc., 110 First St., Suite B Los Altos, California 94022



Special introductory price on this remarkable new course is just $99.50... and if it isn't everything we say it is or even more, return it within 15 days for a full refund!

iasis inc.



Order before Nov. 30, and you'll save a full $25 on the Programmed Learning Course on Microcomputers! In addition, all introductory orders will include a bonus seventh volume, the Microcomputer Applications Handbook!

(After Nov. 30, 1975 price for the complete Iasis course will be $124.50, plus $2.50 for postage and handling.)

Here's my check or money order *(no cash, please). RUSH* my 6-volume Programmed Learning Course on Microcomputers, including the bonus Applications Handbook and programming aids, to the address below. HURRY!

☐ Send one complete course for $99.50 in U. S. funds. My payment is attached. (California residents, please add $5.97 State sales tax.)

☐ Send me information by return mail on quantity discounts.

ALLOW 15 DAYS FOR DELIVERY IN THE U. S. AND 6 WEEKS FOR DELIVERY OUTSIDE THE UNITED STATES.

iasis inc.

Mail today to: Iasis,
110 First St. Suite B
Los Altos, California 94022

You can use your BankAmericard or Master Charge, too!

☐ CHARGE MY ORDER TO THE CREDIT CARD NO. BELOW:

BankAmericard No. ____________

Master Charge No. ____________

For Master Charge, add 4-digit number immediately above your name on the card. It is ____________

HERE'S MY SIGNATURE ____________
(Sign here if credit card charge)

Credit card expiration date ____________

NAME ____________

ADDRESS ____________

ORGANIZATION ____________ MAIL STOP ____________

CITY/STATE/ZIP ____________

An Intel 8080 Op Code Table

Fred Dittrich
312 N 8 St (no 2)
Columbia MO 65201

Fred Dittrich, 312 N 8 St (no 2), Columbia MO 65201, supplies this table of op codes for the 8080 instruction set. The notation for mnemonics is that of the Intel 8080 Microcomputer System Manual (MCS-472-0275), published in January 1975. The table gives all the 8 bit numbers from 0 to 255 in decimal, hexadecimal and octal so the table can also be used as a base conversion chart. The following format is used for the mnemonics:

One byte instructions are shown in capital letters only, such as MOV C, H

Two and three byte instructions have symbols in angle brackets signifying the additional bytes following the op code, such as LDA <LO> <HI>, where <LO> is the low order portion, and <HI> is the high order portion of a 16 bit address; and <DATA> means immediate data; and <DEVICE> = device code

For details of the operation of instructions consult the Intel literature, manufacturers' specifications from 8080 second sources, or the user documentation of computer kits which use the 8080 chip.

Dec	Hex	Octal	Mnemonic	
0	00	000	NOP	
1	01	001	LXI	B<LO><HI>
2	02	002	STAX	B
3	03	003	INX	B
4	04	004	INR	B
5	05	005	DCR	B
6	06	006	MVI	B<DATA>
7	07	007	RLC	
8	08	010		Unimplemented
9	09	011	DAD	B
10	0A	012	LDAX	B
11	0B	013	DCX	B
12	0C	014	INR	C
13	0D	015	DCR	C
14	0E	016	MVI	C<DATA>
15	0F	017	RRC	
16	10	020		Unimplemented
17	11	021	LXI	D<LO><HI>
18	12	022	STAX	D
19	13	023	INX	D
20	14	024	INR	D
21	15	025	DCR	D
22	16	026	MVI	D<DATA>
23	17	027	RAL	
24	18	030		Unimplemented
25	19	031	DAD	D
26	1A	032	LDAX	D
27	1B	033	DCX	D
28	1C	034	INR	E
29	1D	035	DCR	E
30	1E	036	MVI	E<DATA>
31	1F	037	RAR	
32	20	040		Unimplemented
33	21	041	LXI	H<LO><HI>
34	22	042	SHLD	<LO><HI>
35	23	043	INX	H
36	24	044	INR	H
37	25	045	DCR	H
38	26	046	MVI	H<DATA>
39	27	047	DAA	
40	28	050		Unimplemented
41	29	051	DAD	H
42	2A	052	LHLD	<LO><HI>
43	2B	053	DXC	H
44	2C	054	INR	L
45	2D	055	DCR	L
46	2E	056	MVI	L<DATA>
47	2F	057	CMA	
48	30	060		Unimplemented
49	31	061	LXI	SP<LO><HI>
50	32	062	STA	<LO><HI>
51	33	063	INX	SP
52	34	064	INR	M
53	35	065	DCR	M
54	36	066	MVI	M<DATA>
55	37	067	STC	
56	38	070		Unimplemented
57	39	071	DAD	SP
58	3A	072	LDA	<LO><HI>
59	3B	073	DXC	SP
60	3C	074	INR	ACC
61	3D	075	DCR	ACC
62	3E	076	MVI	ACC<DATA>
63	3F	077	CMC	
64	40	100	MOV	B, B
65	41	101	MOV	B, C
66	42	102	MOV	B, D
67	43	103	MOV	B, E
68	44	104	MOV	B, H
69	45	105	MOV	B, L
70	46	106	MOV	B, M
71	47	107	MOV	B, ACC
72	48	110	MOV	C, B
73	49	111	MOV	C, C
74	4A	112	MOV	C, D
75	4B	113	MOV	C, E
76	4C	114	MOV	C, H
77	4D	115	MOV	C, L
78	4E	116	MOV	C, M
79	4F	117	MOV	C, ACC
80	50	120	MOV	D, B
81	51	121	MOV	D, C
82	52	122	MOV	D, D
83	53	123	MOV	D, E
84	54	124	MOV	D, H
85	55	125	MOV	D, L
86	56	126	MOV	D, M
87	57	127	MOV	D, ACC
88	58	130	MOV	E, B
89	59	131	MOV	E, C
90	5A	132	MOV	E, D
91	5B	133	MOV	E, E
92	5C	134	MOV	E, H
93	5D	135	MOV	E, L
94	5E	136	MOV	E, M
95	5F	137	MOV	E, ACC
96	60	140	MOV	H, B
97	61	141	MOV	H, C
98	62	142	MOV	H, D
99	63	143	MOV	H, E
100	64	144	MOV	H, H
101	65	145	MOV	H, L

102	66	146	MOV	H, M
103	67	147	MOV	H, ACC
104	68	150	MOV	L, B
105	69	151	MOV	L, C
106	6A	152	MOV	L, D
107	6B	153	MOV	L, E
108	6C	154	MOV	L, H
109	6D	155	MOV	L, L
110	6E	156	MOV	L, M
111	6F	157	MOV	L, ACC
112	70	160	MOV	M, B
113	71	161	MOV	M, C
114	72	162	MOV	M, D
115	73	163	MOV	M, E
116	74	164	MOV	M, H
117	75	165	MOV	M, L
118	76	166	HLT	
119	77	167	MOV	M, ACC
120	78	170	MOV	ACC, B
121	79	171	MOV	ACC, C
122	7A	172	MOV	ACC, D
123	7B	173	MOV	ACC, E
124	7C	174	MOV	ACC, H
125	7D	175	MOV	ACC, L
126	7E	176	MOV	ACC, M
127	7F	177	MOV	ACC, ACC
128	80	200	ADD	B
129	81	201	ADD	C
130	82	202	ADD	D
131	83	203	ADD	E
132	84	204	ADD	H
133	85	205	ADD	L
134	86	206	ADD	M
135	87	207	ADD	ACC
136	88	210	ADC	B
137	89	211	ADC	C
138	8A	212	ADC	D
139	8B	213	ADC	E
140	8C	214	ADC	H
141	8D	215	ADC	L
142	8E	216	ADC	M
143	8F	217	ADC	ACC
144	90	220	SUB	B
145	91	221	SUB	C
146	92	222	SUB	D
147	93	223	SUB	E
148	94	224	SUB	H
149	95	225	SUB	L
150	96	226	SUB	M
151	97	227	SUB	ACC
152	98	230	SBB	B
153	99	231	SBB	C
154	9A	232	SBB	D
155	9B	233	SBB	E
156	9C	234	SBB	H
157	9D	235	SBB	L
158	9E	236	SBB	M
159	9F	237	SBB	ACC
160	A0	240	ANA	B
161	A1	241	ANA	C
162	A2	242	ANA	D
163	A3	243	ANA	E
164	A4	244	ANA	H
165	A5	245	ANA	L
166	A6	246	ANA	M
167	A7	247	ANA	ACC
168	A8	250	XRA	B
169	A9	251	XRA	C
170	AA	252	XRA	D
171	AB	253	XRA	E
172	AC	254	XRA	H
173	AD	255	XRA	L
174	AE	256	XRA	M
175	AF	257	XRA	ACC
176	B0	260	ORA	B
177	B1	261	ORA	C
178	B2	262	ORA	D
179	B3	263	ORA	E
180	B4	264	ORA	H
181	B5	265	ORA	L
182	B6	266	ORA	M
183	B7	267	ORA	ACC
184	B8	270	CMP	B
185	B9	271	CMP	C
186	BA	272	CMP	D
187	BB	273	CMP	E
188	BC	274	CMP	H
189	BD	275	CMP	L
190	BE	276	CMP	M
191	BF	277	CMP	ACC
192	C0	300	RNZ	
193	C1	301	POP	B
194	C2	302	JNZ	<LO><HI>
195	C3	303	JMP	<LO><HI>
196	C4	304	CNZ	<LO><HI>
197	C5	305	PUSH	B
198	C6	306	ADI	<DATA>
199	C7	307	RST	0
200	C8	310	RZ	
201	C9	311	RET	
202	CA	312	JZ	<LO><HI>
203	CB	313		Unimplemented
204	CC	314	CZ	<LO><HI>
205	CD	315	CALL	<LO><HI>
206	CE	316	ACI	<DATA>
207	CF	317	RST	1
208	D0	320	RNC	
209	D1	321	POP	D
210	D2	322	JNC	<LO><HI>
211	D3	323	OUT	<DEVICE>
212	D4	324	CNC	<LO><HI>
213	D5	325	PUSH	D
214	D6	326	SUI	<DATA>
215	D7	327	RST	2
216	D8	330	RC	
217	D9	331		Unimplemented
218	DA	332	JC	<LO><HI>
219	DB	333	IN	<DEVICE>
220	DC	334	CC	<LO><HI>
221	DD	335		Unimplemented
222	DE	336	SBI	<DATA>
223	DF	337	RST	3
224	E0	340	RPO	
225	E1	341	POP	H
226	E2	342	JPO	<LO><HI>
227	E3	343	XTHL	
228	E4	344	CPO	<LO><HI>
229	E5	345	PUSH	H
230	E6	346	ANI	<DATA>
231	E7	347	RST	4
232	E8	350	RPE	
233	E9	351	PCHL	
234	EA	352	JPE	<LO><HI>
235	EB	353	XCHG	
236	EC	354	CPE	<LO><HI>
237	ED	355		Unimplemented
238	EE	356	XRI	<DATA>
239	EF	357	RST	5
240	F0	360	RP	
241	F1	361	POP	PSW
242	F2	362	JP	<LO><HI>
243	F3	363	DI	
244	F4	364	CP	<LO><HI>
245	F5	365	PUSH	PSW
246	F6	366	ORI	<DATA>
247	F7	367	RST	6
248	F8	370	RM	
249	F9	371	SPHL	
250	FA	372	JM	<LO><HI>
251	FB	373	EI	
252	FC	374	CM	<LO><HI>
253	FD	375		Unimplemented
254	FE	376	CPI	<DATA>
255	FF	377	RST	7

There's More to Blinking Lights Than Meets The Eye

Carl Helmers

A blinking light peripheral is an inexpensive, entertaining addition to your computer system. The use of multiple indicator lamps under computer control to produce moving patterns can lead to many hours of creative programming and pleasant amusement.

Playing with blinking lights is nothing new to people working with computers. Early systems, to say nothing of science fiction movie caricatures, tended to have monstrous front panels with row after row of indicator lamps. These lamps were used to reveal various machine states. As computers became more sophisticated, the need for many of these indicators diminished, but the fascination of making the indicator lamps dance and gyrate in interesting patterns has remained.

As a simple example of moving lights, consider a single 8 bit byte of memory in your computer, which might be called BLINK. Assume that you have also constructed an 8 bit output data latch which drives eight LEDs as shown in figure 1. When a certain program is started, BLINK might be initialized as follows:

0000 1000

Suppose this value is sent to the display, which is set up so that a 1 bit lights an LED, then the display will look like this:

● ● ● ● ○ ● ● ●

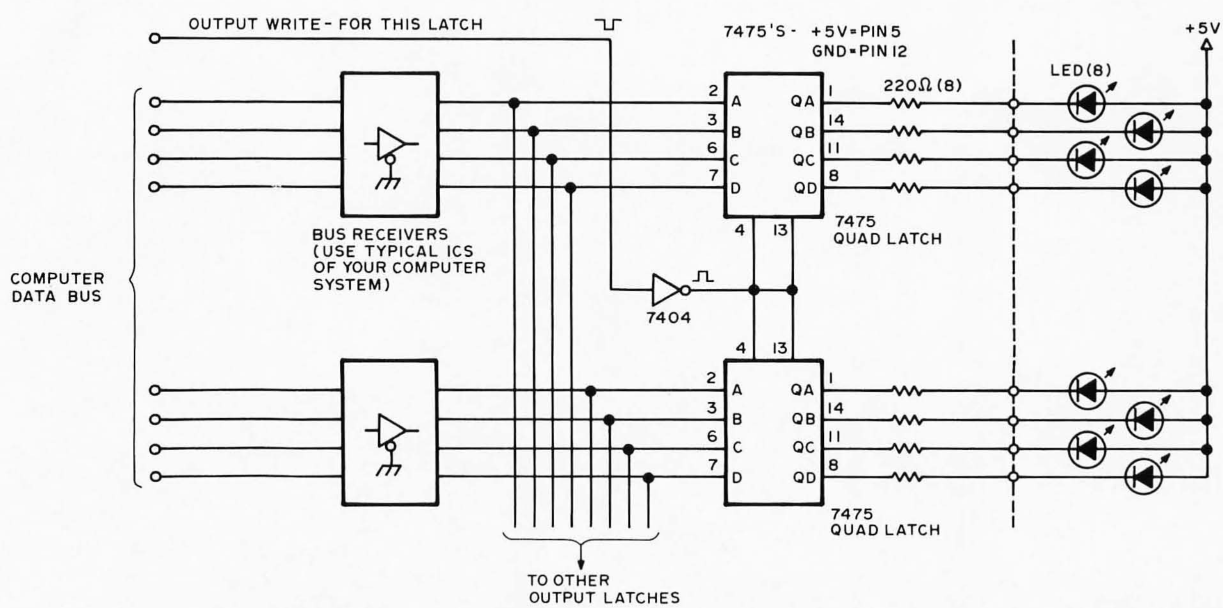


Figure 1: The basic "straightforward" approach to display lamps. One latch is assigned to each of several output ports of the computer. The output port decoding logic will determine when the latch is addressed for output. The result of decoding is a WRITE signal which latches the data presented at the bus receivers. The sample program of figure 2 assumes two such display registers. In principle the idea can be extended to many registers in groups of eight, limited only by the computer's input output addressing capability and the available power for lighting LEDs. In this circuit, TTL fanout limitations on the bus receivers would limit expansion to a total of 10 latches. The LEDs can be mounted on a separate display panel with connections by means of dual in line header plugs and sockets. A neutral tinted glass or plastic cover plate and an attractive wooden frame are good finishing touches for the visible portion of the project.

Figure 2: The CATERPILLAR program listing. Opcodes and addresses are specified in octal notation for the 8008 CPU. Comments at the right are designed to help convert the program to other machines.

```
address   octal code  label    op.   operand     commentary

010/000  250          START    XRA               clear carry and A;
010/001  036 377               LDI   11111111B   first caterpillar word;
010/003  046 000               LEI   00000000B   second caterpillar word;
010/005  304          MARCH    LAE               right byte to A;
010/006  022                   RAL               shift left into carry;
010/007  340                   LEA               then saved for next time;
010/010  303                   LAD               left byte to A;
010/011  022                   RAL               shift left into carry;
010/012  330                   LDA               then saved for next time;
                      *   note that RAL shifts old carry into A each time,
                      *      thus the above accomplishes shift from E's high
                      *      order bit into D's low order bit via carry...;
010/013  177                   OUT   37          write into left lamps;
010/014  304                   LAE               fetch right value;
010/015  175                   OUT   36          write into right lamps;
010/016  026 300               LCI   192D        set delay loop constant;
010/020  307          DELAY    LAM               use
010/021  307                   LAM                 several
010/022  307                   LAM                   longish
010/023  307                   LAM                     instructions
010/024  307                   LAM                       to
010/025  307                   LAM                     s t r e t c h
010/026  307                   LAM                   out
010/027  307                   LAM                 the
010/030  307                   LAM               loop
010/031  307                   LAM               (a BYTE in time saves nine);
010/032  021                   DCC               decrement delay count;
010/033  110 020 010           JFZ   DELAY       if non zero then repeat delay;
010/036  104 005 010           JMP   MARCH       back for another step;
                      *   note that during the operations at addresses 010/013
                      *      to 010/036 the carry bit value set by the last RAL
                      *      is retained unchanged, so that the RAL at 010/006
                      *      will shift the old high order bit of the left register
                      *      (D) into the low order bit of the right register (E);
```

This program uses only the internal CPU registers for its data, and assumes that 8008 output ports 36 and 37 are assigned to latched 8 bit displays so that a visible pattern can be seen.

The key element in creating an illusion of motion is time. If the program starts out with data as shown above, waits a short time, then executes a left rotate instruction, a new pattern will be obtained. The new binary value 0001 0000 can be sent to the display:

● ● ● ○ ● ● ● ●

In the simplest of all motion programs, these three steps are repeated in an endless loop:

1. Rotate BLINK left one bit position.
2. Send BLINK to the display.
3. Wait n milliseconds.
4. Go to step 1.

With a program executing these four steps, the pattern of lit indicators will be seen moving to the left, disappearing on the left in the same step at which it reappears on the right. By changing the program delay (step 3), the speed of the pattern's apparent motion can be changed.

Figure 1 shows two quad latches which are used to drive 8 indicator lamps. The 220 O resistors are typical values for LEDs as indicators. This value allows reasonable brightness with most LEDs. The complement outputs of the 7475 ICs used as latches produce a lit LED for each 1 bit received. When you purchase LEDs for a blinking light

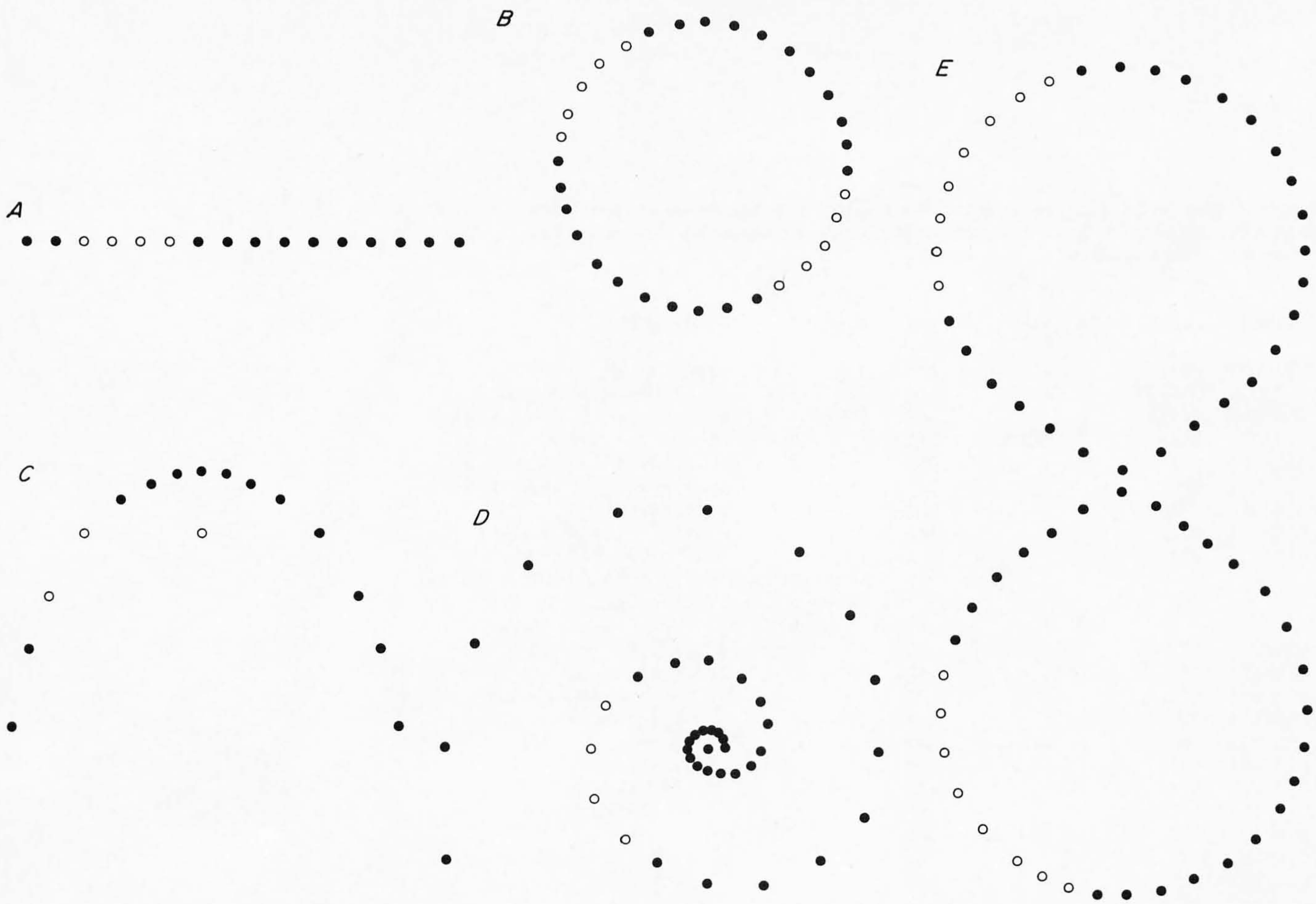


Figure 3: Creativity in the arrangement of the lamps of your blinking light peripheral can make the show more impressive and interesting. A: The traditional blinking light display, a la control panel, is a straight line of sixteen lamps in a row using two output latches. B: Tradition is nice, but how about a bit of circular thinking, using thirty two lamps in a circle with four output latches. C: Don't rule out the bouncing ball effect either. Here we make a parabola shape using sixteen lamps, one at the focus, with two output latches. D: With enough lamps, we can make a spiral show with one or more arms. Here are forty lamps using five output latches. E: There are an infinite number of patterns to be made. In this case, infinity consists of sixty four lamps using 8 output latches.

display, make sure they are all the same, as the display will not look as attractive if lamps of different types are mixed.

Now, suppose you use four latches for 16 bits of data and 16 LEDs. How do you program an 8 bit computer to do the shifting light pattern function for 16 bits? The basic way in which this is accomplished is to shift the bits through the carry flag of your machine.

Figure 2 illustrates a program which does a 16 bit shift in an 8008 microcomputer, and sends the data to the output ports reached by the OUT36 and OUT37 instructions (octal 175 and 177, respectively). The result is a moving display of 16 lights. A band of several bits is always marching right to left around the display at a steady rate.

But, why let the imagination end at a mere 16 bit display? The use of IO ports can be extended without too much cost (given limitations on power supplies, of course). The cost of two 7475s, eight resistors, and eight LEDs is about $3 (using BYTE's advertising pages as a source of prices). Thus a very reasonable display of 64 LEDs will set you back only $24 and the time it takes to put it together.

Using a bit of imagination, the extension of the program can lead to interesting patterns running around non-linear configurations such as those illustrated in figure 3. Variations in the patterns result in displays of light. Beyond the scope of this short article are more complicated programs using the same display peripheral: Programs in which the light patterns do more interesting things than simply chasing around the racetrack at uniform speeds.

Try building a simple 16 bit blinking light display version first, then go on to bigger and better things. Then see if you don't agree that there's more to BLINKing lights than meets the eye! ■

Photographic Notes on Wire Wrapping

Carl Helmers

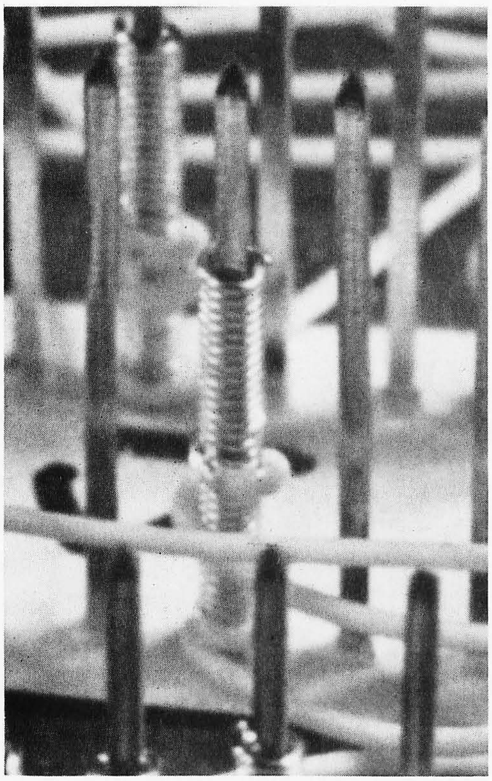


Photo 1: One of the most widely used custom assembly methods in the computer industry is wire wrapping. Special square wire wrap posts are built into sockets or mounted directly onto components. The stripped ends of insulated wires are connected to the posts using a special electric or hand operated tool. The result is an excellent connection with gas tight "vacuum welds" at each corner of the post as the sharp edge cuts slightly into the wire. Wire wrapping is a quick, easy and permanent way to connect point A to point B in your custom circuitry.

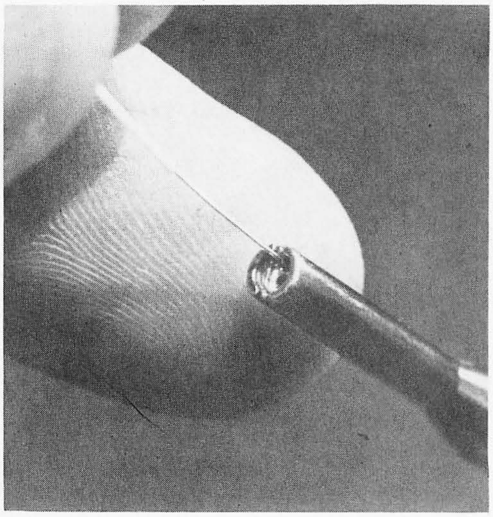


Photo 2: The business end of an industrial quality wire wrap tool is the bit and sleeve assembly. The bit is driven by a high speed motor (or hand operated mechanism) and rotates around the post to be wrapped. The center of the bit has an alignment hole which fits over the wire wrap post to guide the operation. The bit also has a groove along its length into which the stripped end of a wire is inserted as shown. The wire is held in the groove by the sleeve and is pulled out during the process of wrapping. The sleeve itself does not rotate and is held rigidly in position.

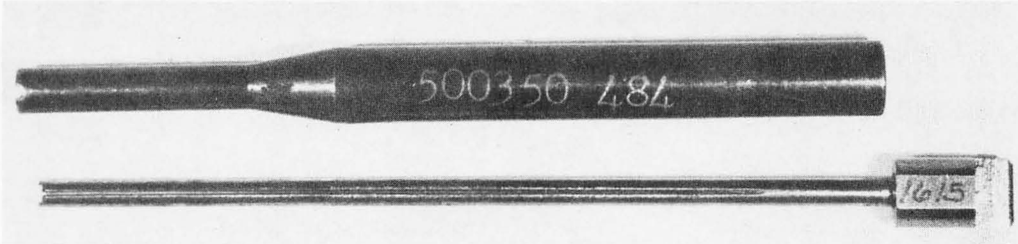


Photo 3: The bit and sleeve may be removed from the typical industrial wrapping tool. The groove for the wire can be seen as a line running the length of the bit in this photograph. Note how the groove has an enlarged section near the tip of the bit. This enlarged section allows a small length of unstripped wire to be inserted so that one turn of insulation will be wrapped around the post.

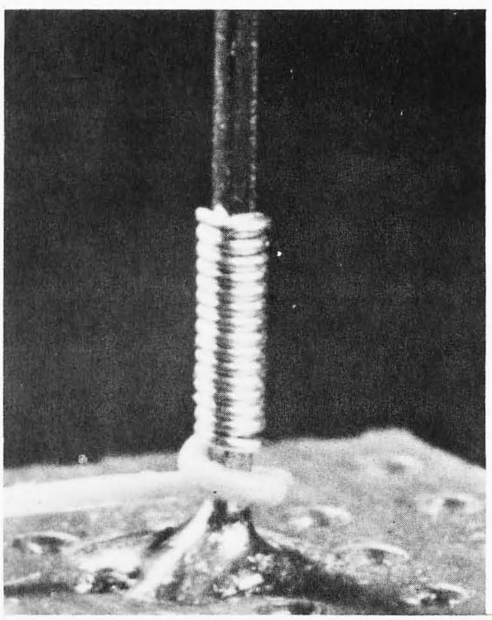


Photo 4: When the trigger has been pulled on a loaded wire wrap gun, the result (hopefully) will be a perfect wrap every time. The wrapping action begins at the bottom with a single turn of insulated wire followed by successive layers of stripped wire. In this picture you can count 15 layers of bare wire on the side facing the camera. The single turn of insulated wire is produced by what is called a modified wrap bit. This single turn around the post uses the insulation to help act as a strain relief to prevent breakage when handling wires while debugging a system. A standard wrap bit would start the first coil of bare wire minus the strain relief feature.

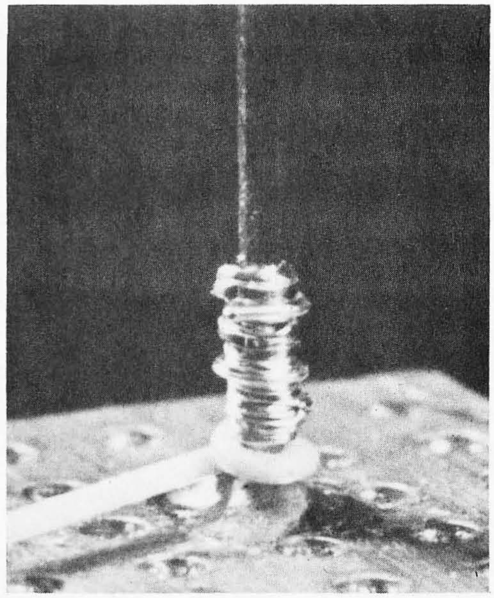


Photo 5: To quote the previous caption, "When the trigger has been pulled on a loaded wire wrap gun, the result (hopefully) will be a perfect wrap every time." Here is an example where hope was in vain. A problem which sometimes occurs is the "wrat's nest" wrap shown in this photograph. The normal operation of the wrapping gun includes retraction of the bit inside the sleeve as the wire layers are added to the coil around the post. A spring loaded mechanism in the gun provides for this controlled retraction. Occasionally the bit sticks and does not retract properly. This results in a jumbled mess as layers of wire wrap over previously wrapped layers. If the wrat's nest wrap occurs only occasionally, the likely cause is your tool position while wrapping. If the bit is not centered about the wrapping post, friction in the pilot hole can prevent the bit from retracting properly. If the wrat's nest wrap occurs consistently a likely cause is dirt and grime buildup on the bit. If particles of metal scraped off the wire are allowed to accumulate, the resulting friction can inhibit retraction of the bit. Thus if the gun consistently makes wrat's nest wraps, removing the bit and cleaning it may resolve the problem. After cleaning, the bit may be oiled. Oiling should be done very sparingly at the gun end of the bit so that excess oil does not get in the groove and onto the wire. When a wrat's nest occurs, whatever the cause, good practice indicates that you should remove the mess and start over.

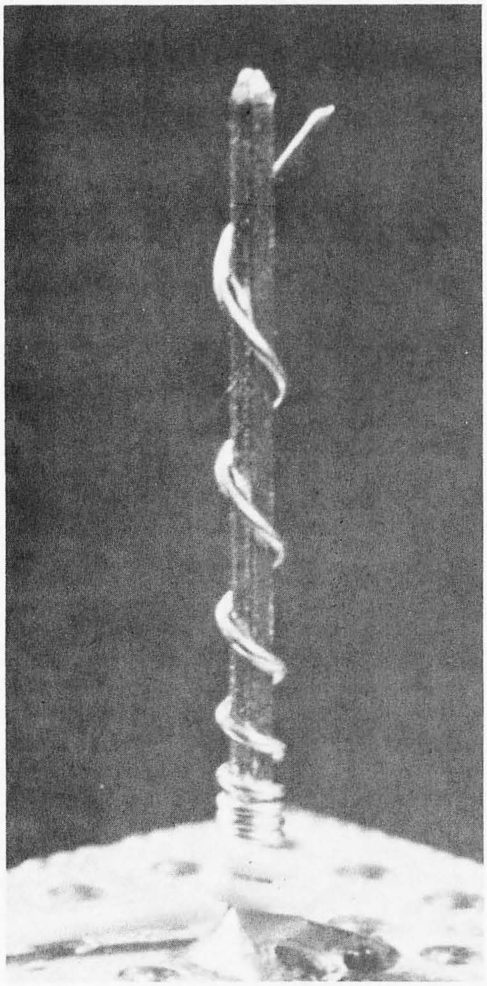


Photo 6: Continuing on the theme of less than optimal wrapping results, this picture shows an example of the "impatient wrap." The primary cause of the impatient wrap is your own impatience to see the result. If the wire wrap gun is removed from the post while the wrapping action is under way, the result will be widely spaced coils of wire often covering the length of the post. This example is an exaggerated version concocted on purpose; the typical case is a normal base to the coil with the top one or two layers spread out. The solution to this problem is to allow the gun to do its work before removing it from the post. In a case such as the one shown here, removal of the connection and rewrapping should be done. In milder forms of the impatient wrap, the wrap would not necessarily have to be replaced.

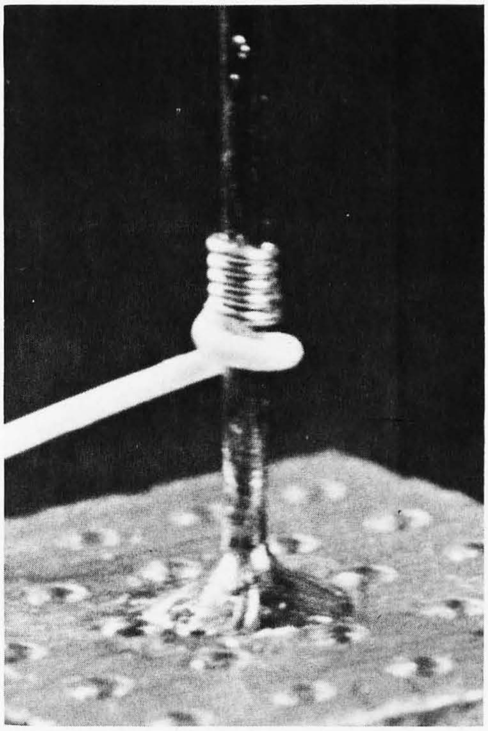


Photo 7: The length of the coil produced by the wrapping action is a function of the length of bare wire prepared for insertion in the tool. For the typical .025 inch square wrapping posts with 30 gauge wire, approximately 8 layers will be added to the coil for each inch of bare wire inserted into the bit. This photo shows an example of the "pigmy wrap." When a short coil of wire is produced, one possible explanation is that not enough wire was stripped when preparing for the connection. (Another possible explanation is that the wire broke, leaving most of the stripped section inside the groove where it must be removed by disassembling the tool.)

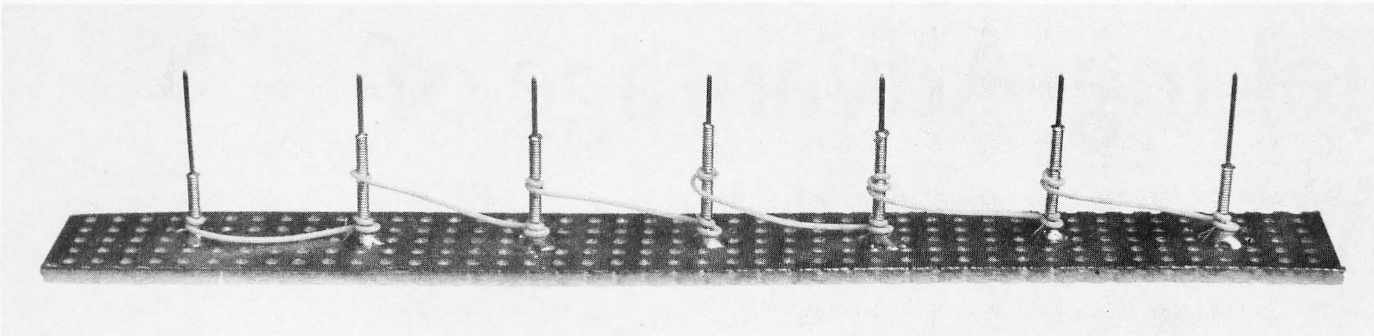


Photo 8: It is often necessary to connect several wire wrap posts to one electrical bus. The method of accomplishing this is to create a chain of point to point wraps. In such a chain each wrapping post except the first and the last will receive two wraps: a level one wrap at the bottom and a level two wrap above it. One way to build the chain is to start at one end connecting one post to the next until the last connection is reached. In the model shown here, wrapping began at the right and progressed toward the left. There is a major disadvantage with this technique of chaining. Suppose you want to remove the link between the two posts furthest to the right in the photo. Note that in order to remove the first level wrap on the second post from the right, you must first remove the second level wrap on that post. However, the wire to the second level comes from the first level wrap of the third post and must be removed at both ends since wire wrap wire cannot be recycled. But this presents the same problem all over again at the third post, and the problem will propagate down the chain until all the wire has been removed. One way to get around the problem simply is to clip the removed wire and leave a dummy wrap on the post, making any new connection at the third level if the post is long enough. There is an alternative method of connection shown in the next photograph which minimizes the effects of a chain change on long chains.

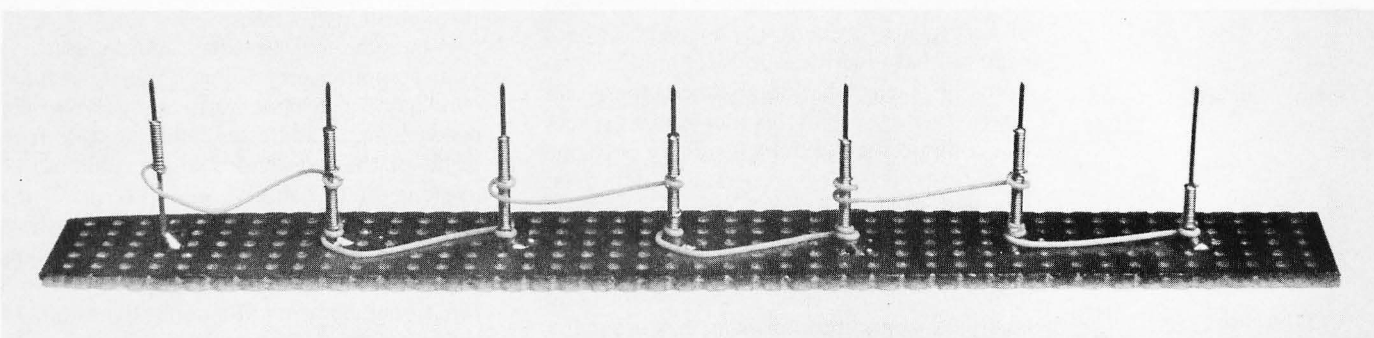


Photo 9: The alternative method of chaining which minimizes propagation of changes is illustrated in this photograph. This technique is highly recommended for initial wiring of any bus which connects more than 4 posts. The principle which governs this form of wiring is simple: Wires are connected to the same wrap level at each end. If a change is required at level two, only one wire has to be removed to break a link. If a change is required at a level one link, three wires must be removed (and two possibly rewired). When wiring a bus this way, the first step is to connect all the possible pairs of posts in the bus with wraps at level one. For buses with an even number of posts, this will put a level one wrap at every post; for buses with an odd number of posts, one post will be left without a connection after all the pairs are wired. Then, each end of each pair is connected to its neighboring pair with a level two wrap, completing the chain. For odd length buses, the solitary unwrapped post is connected to one of the pairs at this stage. This method of wrapping works only with the initial wiring of a chain. If later changes are made, such as adding or deleting segments in the middle, necessity may force a hybrid version of the two techniques of chaining.

Taking Advantage of Memory Address Space

James Luscher
2012 Jefferson Street
Madison WI 53711

A computer's address space is not necessarily equivalent to the amount of memory it has installed.

The address space of a computer is the set of addressable data locations provided in the design of its instruction set. A common misconception is to view a computer's address space as the equivalent of the actual memory that the machine has. This point of view is both incorrect and too restrictive for the hobbyist or engineer who is able to modify his machine.

Thinking in terms of address spaces is more productive than thinking in terms of memory for three reasons: First memory is normally installed in blocks of addresses where each word is identical in length, speed of access, etc. In contrast, an address space need not have identical physical implementations at each location. Second there are often several logically distinct address spaces in a computer while there is usually only one main memory for programs and data. Multiple address spaces have separate functions or functionally distinct address spaces (e.g., input output device space and main memory address space) can be merged. Lastly, a computer normally has a smaller main memory than it is capable of addressing, thus wasting some (or most) of the address space.

Non-Memory Address Spaces

As an example of a non-memory address space, let's first consider the ADr instruction (add register to accumulator) as implemented on Intel's 8008 microcomputer chip. Its bit pattern is 10000RRR, where RRR represents the register address field. This defines an address space for the 8008 CPU which is not directly associated with main memory at all. As a second example, consider the 8008 input instruction, INP. The bit pattern for this instruction is 0100MMM1 where MMM is the input unit address field. This defines a second address space for the 8008 which is also unrelated to main memory addressing. Thus there are several distinct address spaces associated with the 8008 design.

The designers of a computer have to give careful consideration to all the address spaces supported by the computer they create. The input instruction address space allows up to 8 locations as input ports. If the designer of the Intel 8008 had wanted to allow 16 input devices he would have needed 4 bits. Using the larger address space means that some instructions must be modified, eliminated or restricted because the bit retained for addressing cannot be used to distinguish instructions.

You pay for the extra address space in your computer even if you don't use it fully. You pay for 8 input ports even if you only need 2 input addresses, and the cost is in terms of the instructions which could not be implemented. You also pay in terms of the bits of memory which are required to hold the unused parts of the addresses. Deciding how to handle the addressing of various functional elements in a computer is one of the most crucial issues in the design of an instruction set.

Non-Uniform Address Spaces

To see that an address space need not identify uniform arrays like main memory let's re-examine the 8008's ADr instruction. The register address space can specify one of 8 locations with 3 bits. However there are only 7 registers in the 8008. Each of the registers numbered 0 to 6 specifies a byte of data which is part of the 8008 CPU. These registers, named A, B, C, D, E, H and L, are the seven "normal memory" addresses in the 8008 register address space. Each of these registers, if selected by the RRR bits of the ADr instruction can have its content added to the A register when the instruction is executed. The eighth address in this address space (numbered 7) is different in function

from the other seven. Address seven in the register address space is not a register at all. Referencing register 7 causes the H and L registers to be placed end to end forming a 16 bit address register. This is then used to address a byte in main memory. Thus one element of the register address space of the 8008 is very different from the other seven elements (see figure 1).

Using Non-Uniform Main Memory Address Space

The previous example illustrated a non-uniform address space built into the architecture of a specific computer chip, the 8008. There is no reason why the hobbyist can't use the same concept of a non-uniform address space element to his advantage in the design of custom systems or additions to existing systems. Suppose that someone has a 6800 processor with 4096 bytes of memory and wants to implement the LIFE application on this machine. With a 64 x 64 grid, 4096 bytes of memory alone would be required for data storage if one byte is used for each element of the LIFE matrix. Two normal solutions to the problem are to buy more memory or to implement subroutines which store 8 matrix elements into each byte, thus requiring only 512 bytes for data plus storage for the packing and unpacking subroutines. This last approach has the unfortunate drawback of being slow, not to mention its added programming difficulty.

There is another clever way to solve the problem of accessing large arrays of bits, a way which is open to the hobbyist who designs his own machine or doesn't mind adding to his machine. It is possible to build a 4096 bit memory at a cost much lower than the cost of a full 4096 byte memory board. In effect you are replacing the packing and unpacking routines with the chip address selection hardware at a nominal dollar cost and a great increase in speed. Figure 2a illustrates the concept of the 4096 bit memory added to a system which already has 4096 bytes of random access memory.

Since only one bit of information is involved, the way the addressed bit is hooked up to the bus interface data lines is a matter of personal preference and programming convenience. One way (figure 2b) is to connect the single active bit to the low order bus line at the bus interface. Testing the state of the stored bit could then be done by loading from the memory location of the bit and testing the zero flag with a conditional branch instruction for the computer being used. A second way is to connect the active memory chip to the high order bit, in which case the sign of the result (in two's complement notation) corresponds to the present state of the bit in memory. In either of these first two ways, the unused bits may be wired to logical zero or one. A third way, shown in figure 2d, is to connect the output of the single bit memory to all 8 data lines at the interface to the bus. When this method is chosen, the data value loaded from the one-bit memory locations will be either zero or -1 (hexadecimal 00 or FF). The one bit wide memory is much less expensive since only the single bit being used at any address has a physical implementation as a memory device.

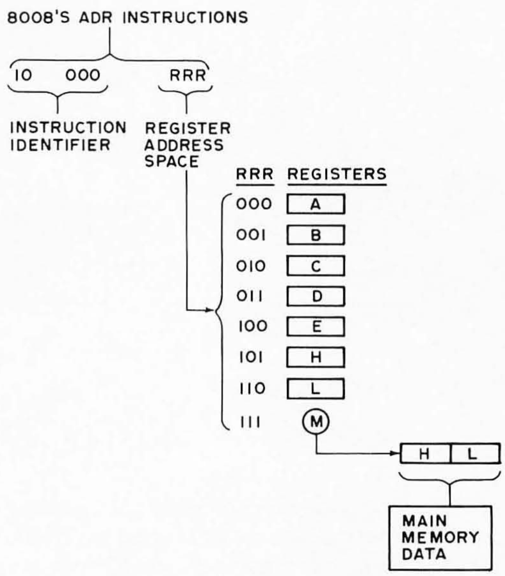


Figure 1: An example of a small non-uniform address space is provided by the register address space of an 8008 microprocessor. Seven out of eight possible locations are registers of the processor; the eighth (M) however has a very special hardware meaning and is not implemented on the chip at all.

Designers of computer architectures have to give careful consideration to all the address space supported by the computer they create.

A 6800 system which uses only 4096 memory addresses is not using 94 percent of its total available address space of 65536 possible locations. There is plenty of space for another 4096 single bit memory. This approach is cost effective, speed effective and makes better use of the hardware resources inherent in the CPU chip design. It is a good solution to the problem of storing an array of bits and can be used in any system which does not require byte memory or peripherals at every available location in address space.

IO and Other Uses of Address Space

In the article "Notes on Parallel Output Interfaces" in BYTE #3 (page 52), Carl Helmers pointed out how memory address space can be used to address output devices. In such a usage, the IO address space and the memory address space of a more conventional design are one and the same thing. This is the only way IO is done in the DEC

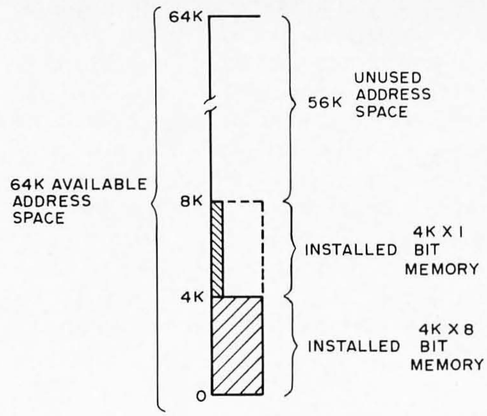
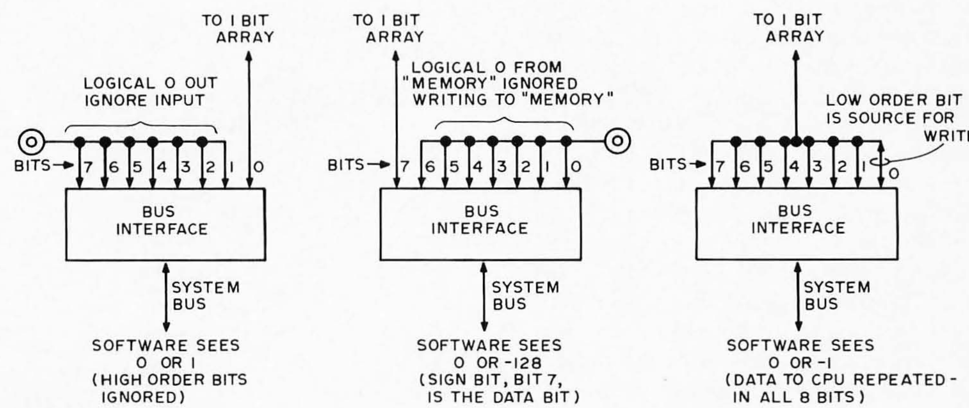


Figure 2: For purposes of manipulating arrays of single bit data, it is possible to use otherwise empty memory address space for a partial implementation of read-write memory. a. illustrates the memory allocations with such a partial array in a typical example. b., c. and d. show three alternative connections of the partial word's data to the bus interface.

Don't make the mistake of thinking you have to include every bit of every word in a memory. Leaving out one or more memory chips will not in general affect the function of the remaining portions of the memory module.



PDP-11 family of minicomputers, as well as microcomputers like the 6800 and the 6501 family. These machines have no IO instructions at all. In the PDP-11 family, there is one particular 4096 word block of memory address space reserved for use with external devices; for most microcomputers, specific allocations of IO addresses in memory address space are a discretionary choice of the system designer.

The addressing concepts used to interface arbitrary IO devices can also be used to add new functions to your computer. Since a CPU cannot tell what kind of memory it is interacting with over the bus, it is possible to have memory locations which serve special computing purposes. Suppose that you want to manipulate 16 bit words with a 6800 system. Specially implemented address space locations can be used to store and manipulate data with the longer word length independent of the CPU's restrictions.

Figure 3 shows two 16 bit registers and one control register for a simple 16 bit arithmetic unit. Registers A16 and B16 occupy two memory address space locations each, for a total of four locations. The control register occupies one address location. Now suppose that locations X and X+1 on the main memory section of the computer contain a 16 bit number, and that locations Y and Y+1 contain a second 16 bit number. A routine to add X to Y and store the result into Y could be implemented fairly simply:

```
LDX   X      fetch first operand (16 bits);
STX   A16    store into 16 bit A register;
LDX   Y      fetch second operand (16 bits);
STX   B16    store into 16 bit B register;
LDAA  #$01   define control code
STAA  C16       and store it into external unit;
LDX   A16    fetch accumulated result;
STX   Y      and store it into Y;
```

The addresses A16 and B16 refer to the registers of the external math unit built along the lines of figure 3; the control register C16 when set with an appropriate code will execute the math operation — which could just as easily have been a multiply or a divide or some other function such as block data movement or an array operation implemented in hardware. What we have done here is to augment the data handling capabilities of the microprocessor by adding hardware which interfaces through a series of locations in memory address space. With this type of hardware addition, the only programming required is thus a series of loads and stores needed to move the data to and from the special memory locations, thus saving a lot of microprocessor programming and making extended precision arithmetic much faster than the equivalent software version. While the circuitry to perform such extended

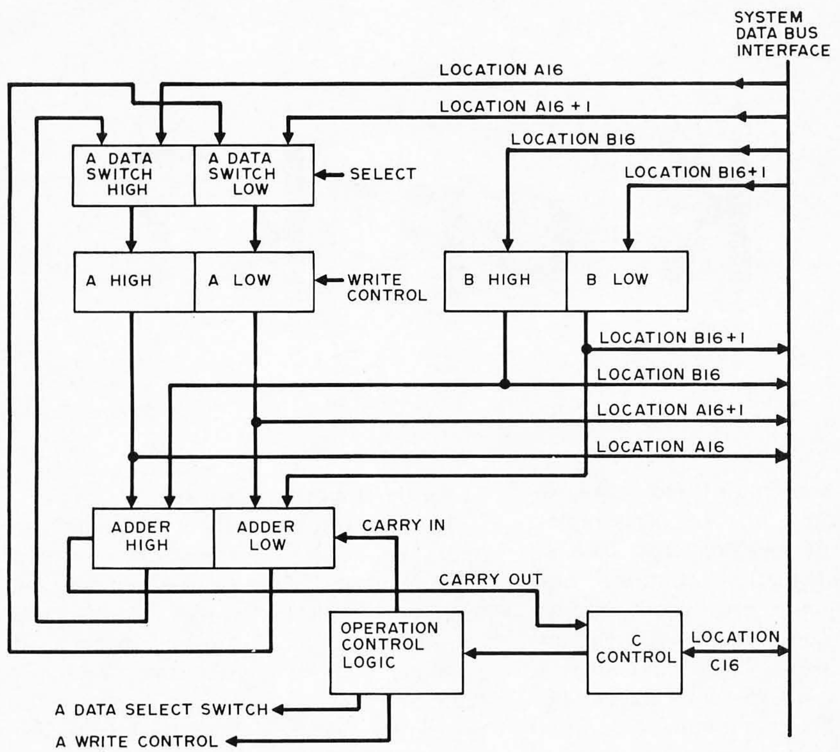


Figure 3: Another use of memory address space resources is to interface special function hardware. This diagram shows the concept of a 16 bit arithmetic module which can be constructed and interfaced as four data locations and one control location in memory address space.

operations is not trivial, it is a lot less expensive and easier to accomplish than building a complete CPU with instruction decoding, conditional branching, subroutine calling mechanisms, interrupts and all the other capabilities needed in a processor. This is the way to build an "extended instruction set" in a machine which nominally does not have any extension possibilities.

. . . the central theme is to always keep an open and creative outlook towards utilizing all the capabilities your machine provides . . .

Extended arithmetic hardware of this kind is very common on medium and large scale machines where speed improvements are needed. For example in floating point arithmetic which involves a lot of double precision addition and shifting, a common commercial technique is to buy a fast floating point processor from an outside vendor and plug it into a minicomputer. This type of extension of the central processor capabilities by doing computation and data manipulation in the address space (external to the CPU proper) is something that the advanced hobbyist may find to be a challenging and useful addition to the system. Whether adding instruction set extensions or implementing partial memory, the central theme of this is to always keep an open and creative outlook towards utilizing all of the capabilities your machine provides – especially the address space. ■

K or k?

Manfred Peshka

The lack of conventions and standards used to abbreviate units of measure, to determine scale of measurements, base of numbers, or mathematical, relational, and logical operators, may occasionally lead to confusion and misunderstandings. There are people who prefer to use such expressions as KB for 1024 bytes, CPS for characters per second, LPM for lines per minute, etc., while others follow different conventions, such as kHz for 1000 hertz, m/s^2 for meter per squared second, MB for megabyte (either 1,000,000 or 1,048,576 bytes, depending upon convention), and so on. Some people like to use AMPS, or Amps, or amps, or A, to denote the unit to measure electric current. We find abbreviations varying depending upon scale; sec stands for second, while ms means millisecond. Similar problems may arise from the use of numerals to indicate numbers: 11 may mean in binary the value three, in octal the value nine, in decimal the value eleven, and in hexadecimal the value seventeen. Also relational, mathematical, and logical operators are not consistently used. Some writers prefer to use the symbol AND, others the ampersand (&), and others use the dot (•) to indicate the logical AND operation. You find the slash (/) used for division of numbers, or to mean the logical OR operation.

To help our readers understand the use of these different symbols, BYTE will attempt to consistently employ certain abbreviations for frequently used units of measure, scale factors in measurements, operators, and number bases.

All numbers are assumed to be decimal numbers unless it has been stated specifically to the contrary: In these instances the base of the number system is indicated in the text, in a column heading, etc. Leading zeros are omitted. The hexadecimal number 0F will be written as F for the same reason that leading zeros in decimal numbers are omitted. We generally do not write 015 but 15 for the decimal number fifteen.

Leading zeros, on the other hand, are important in the representation of strings. Arithmetic operations require numbers, and logic operations use strings. In the latter case, leading zeros are significant. For example, a 16 bit register is to be set to a value of 15. One approach, using numbers, is to zero the register and add the hexadecimal number F to the register. Another approach, using strings, is to load the register with the bit string '0000 0000 0000 1111' which can be represented in hexadecimal notation as '00 0F'. The spaces within the string quotes are not part of the symbols representing the string value; they simply add to the legibility and may be omitted. In the above example, the strings were qualified: The first one is a bit string, and the second one a hexadecimal string. An unqualified string, therefore, is assumed to be a text string consisting of letters, digits, and punctuation marks including spaces: ' ' is a string of a single space on your typewriter or terminal which translates to '40' hexadecimal or '0100 0000' binary in the Extended Binary Decimal Interchange Code (EBCDIC).

In other words, numbers are decimal, and strings, which are to be enclosed in string quotes, are the usual letters, digits, and special symbols, unless they are specifically qualified as binary, octal, or hexadecimal representations.

Ambiguity exists also in the symbols k and K used as scale factors. According to Webster's New Collegiate Dictionary (1975, page 629), the lower case letter k means 1024; according to conventional use, such as kg for kilogram, km for kilometer, etc., this symbol means 1000. In the computing literature, on the other hand, the capital letter K is often used in measuring memory size and line transmission speeds. In these instances K means a factor of 1024. BYTE will follow this convention: K means 1024 and k is a factor of 1000. There are many more scale

factors used in scaling measurements. Table 1 shows the various unit of measure prefixes which will be used in BYTE.

All but one scale factor are powers of ten. The exception is K which equals 2**10 or 1024. The two asterisks have been employed to indicate exponentiation instead of the frequently found superscript 2^{10}. The purpose of this notation is to facilitate the printing of formulas on line printers and terminals which otherwise would require special hardware and programming to raise the exponent. In a similar manner the slash (/) is being used to indicate division. Table 2 below lists operations performed on numbers.

Table 1: Scale of Measurement Prefixes

Abbreviation	Meaning and Example
T	Tera or one trillion; five terabits or 5 Tb
G	Giga or one billion; two gigahertz or 2 GHz
M	Mega or one million; 28 megabytes or 28 MB
K	K or 1024; two K bytes or 2 KB or 2048 B
k	Kilo or one thousand; two kilohertz or 2 kHz
h	Hecto or one hundred; 5 hectometers or 5 hm
D	Deka or ten; six dekaliters or 6 Dl
d	Deci or one tenth; five decimeters or 5 dm
c	Centi or one hundredth; 90 centimeters or 90 cm
m	Milli or one thousandth; one millihenry or 1 mH
u	Micro or one millionth; six microfarads or 6 uF
n	Nano or one billionth; 18 nanoseconds or 18 ns
p	Pico or one trillionth; four picofarads or 4 pF

Table 2: Mathematical Operations

Abbr	Meaning and Example
x ( )	Function, where x is replaced by a function abbreviation, and the arguments are placed in parentheses, e.g., MAX(4, 7, 2) is 7, the maximum of the three values.
**	Exponentiation; 2**10
–	Negation, a prefix operator; – – 4 is 4
*	Multiplication; A * 4
/	Division; 4/8
+	Addition; A + C
–	Subtraction; 5 – 4

Table 3: Concatenation and Comparisons

Abbreviation		Meaning and Example
CAT or	||	Concatenation; 'aC' CAT ' ' yields 'aC '
LT	<	Less than; ' ' LT 'a' in EBCDIC
NL	¬<	Not less than
LE	<=	Less than or equal to
EQ	=	Equal to
NE	¬=	Not equal to
GE	>=	Greater than or equal to; if A GE B then . . .
GT	>	Greater than; it is false that 5 GT 6
NG or	¬>	Not greater than; 5 NG 6 is true

Besides mathematical operations which require numbers, relational operations (comparisons) may be performed on either numbers or strings; in addition, only strings may be concatenated, i.e., tied together. The string 'abc' concatenated with '1234' results in the string 'abc1234'. This may be written either 'abc' CAT '1234' or 'abc' || '1234'. Since not all terminals have these special symbols, an alternative set of two or three upper case letters can be employed. The abbreviations for relational and concatenation operations are shown in table 3.

While the concatenation of two strings results in a string, a comparison operation returns a value of true or false which can be expressed as a bit. A binary '1' represents true (or yes) and '0' false (or no). This value of a comparison may be assigned to a logical (bit string) variable. In this context, the different uses of the equality symbol become very clear: It is used to assign a value to a variable, and to test equality.

In order to make the difference between the assignment operation and the equality comparison more clear, we will use the colon followed by an equal sign (:=) to denote the assignment, and reserve the equal sign for comparisons of equality. To set a 16 bit register RA to a value of 15, you may write RA:=15; to compare the register with the value 15, you may write RA=15. In this instance the comparison happens to return a true value, i.e., a '1' bit which may be saved in a bit string variable named SWITCH by writing SWITCH:=RA=15. Of course, this latter expression does not mean that you may type it into your computer and expect results. This depends on the availability of a compiler which would generate the appropriate code. On the other hand, this statement expresses an intent and a purpose of a computation, no matter how many machine instructions need to be issued, regardless of the computer at hand.

Like numbers, bit strings have their own set of operations resulting in bit strings. Special graphics are often used to indicate these operations. For example, some people prefer the dot (·) to indicate logical AND, others use the ampersand (&), and others use alphabetic keywords like *and* or AND. We will try to be consistent in our use of these symbols in order to facilitate your reading of the magazine. Since only a few computer output devices have extended graphic character sets, two choices are presented in table 4.

Abbreviations of units of measurement vary widely. We find, for example, amp, amps, AMP, Amp, Amps, AMPs, AMPS, ampere, amperes, a or A to denote the unit of measure for electric current. Often abbreviations are not used consistently, e.g., sec versus ms which stand for second and millisecond, respectively.

There also seems to be little consistency in spelling and hyphenating, e.g., Hertz or hertz, Giga-Hertz, GigaHertz or gigahertz, etc. We decided, therefore, to spell units of measure in lower case letters. After all, these are measurement units and not the people after whom they were named.

The inconsistent usage of abbreviations creates a real problem; it's like learning different languages, or dialects, to say the least. We thought that frequently used units of measure should be abbreviated as short as possible, preferably as a single letter, unless this conflicts with standard usage. We will attempt to use capital letters for abbreviations because they stand out more clearly than lower case letters.

Acronyms, by the way, are always capital letters because each letter represents the first letter of each word of the compound term for which the acronym stands. Multiple letter abbreviations, on the other hand, may have an initial capital letter followed by lower case letters, such as Hz for hertz. Table 5 lists some of the more frequently encountered units of measure in BYTE. ■

Table 4: Logical Operations and Assignment

<table>
<tr><th>Abbreviation</th><th>Meaning and Example</th></tr>
<tr><td>NOT or ¬</td><td>Denial, a prefix operator; '0' is ¬ '1' is always true</td></tr>
<tr><td>AND or &</td><td>Conjunction; C := A & B</td></tr>
<tr><td>NAND</td><td>Negated conjunction</td></tr>
<tr><td>OR or |</td><td>Alternation D := C A</td></tr>
<tr><td>NOR</td><td>Negated alternation</td></tr>
<tr><td>:=</td><td>Assignment; X := 15</td></tr>
</table>

Table 5: Units of Measure Abbreviations

Abbreviation	Meaning and Example
A	ampere; 5 kA, five kiloamperes
b	bit; 1 Kb, one K bits, K bits
B	byte; 14 KB, 14 K bytes
C	character; 14.8 C/s, 14.8 characters a second
db	decibel; 40 db, 40 decibels
g	gram; 3 Dg, three dekagrams
H	henry; 50 mH, 50 millihenry
Hz	hertz; 2 GHz, two gigahertz
m	meter; 10 m, ten meters, ten meter band
O	ohm; 28 kO, 28 kiloohms
s	second; 1 ns, one nanosecond
V	volt; 16 mV, 16 millivolts
W	watt; 2 MW, two megawatts

For The Joules, It's a Steal

I just bought a 105 W power supply for under fifty cents a Watt. This is the "sophisticated logic supply,"* 21 A at 5 V which is available from at least two sources: Meshna Electronics, P.O. Box 62, E Lynn MA 01904 and Electro Science Mart, 119 Foster St., Peabody MA 01960. Using standard TTL logic, a good rule of thumb is to allow one A at 5 V for approximately 30 ICs. Thus this power supply, manufactured by Dressen-Barnes, is capable of powering approximately 600 ICs. For the average computer hobbyist who is designing his or her own system, one of these supplies will be perfect for the main power supply. It is cheap enough to buy right away, and it has sufficient capacity for just about any conceivable future expansion (not being "average" myself, I just ordered a second supply to cover my anticipated future needs).

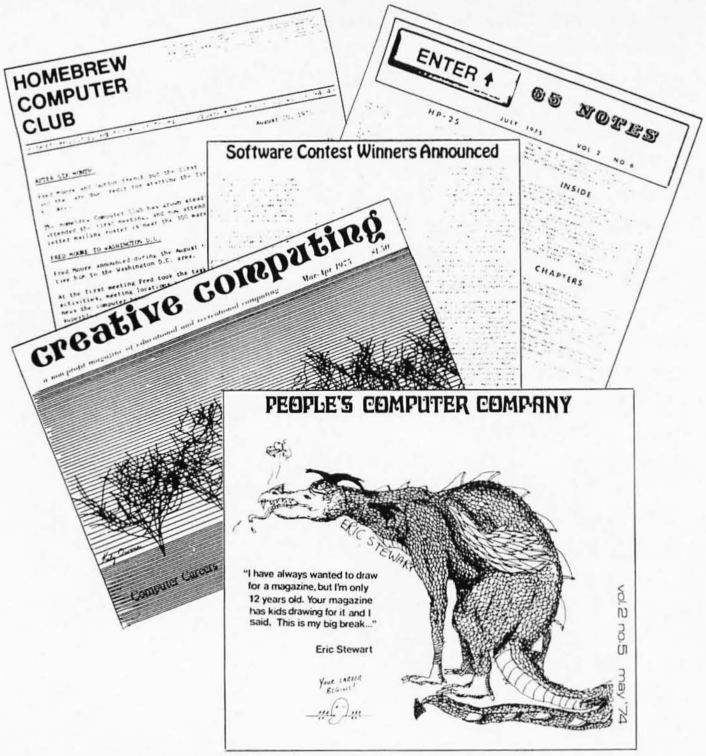
The supply comes as a black box with several interconnection terminals. Two of these terminals are for the 110 V AC input. Three other terminals are for the positive output, negative output, and ground. For +5 Volt TTL then, the negative output should be connected to ground. Since it is a black box, the supply does not come with line cord, pilot light or switch. I surmounted this problem by building the supply into an aluminum chassis along with a custom ±12 V power supply. Although the unit is built with massive heat sinks, I decided to mount a ventilating fan to ensure good air flow for an extra measure of cooling.

The power supply comes with a fuse protected output and a voltage adjustment pot. The pot adjusts the output from 4 V to about 5.6 V. I also hooked up a test load of a ½ O, 50 W resistor to check on the ripple of the supply with a 10 A load. I could see only about 10 to 15 mV of ripple on my oscilloscope. Also, voltage stability over a period of 24 hours was good; I saw no measurable drift.

In summary, this supply has the capacity to rectify a lot of current problems. Watt a bargain! ■

*As advertised on page 95 of the November 1975 BYTE.

Peter H. Helmers
79 Evangeline St.
Rochester NY 14619



Clubs and Newsletters

Pittsburgh Group

The following note was received from Eric Liber and Fred Kitman of Pittsburgh:

Dear Fellow Computer Phreaks:

Just a quick note to let you know that hobby computing and computers are alive and well in Pittsburgh. The name of the organization is the Pittsburgh Area Computer Club. We had our first meeting on October 8, 1975 and even though the weather was inclement we had 15 people in attendance. In addition there is definite interest from at least that many more! Not bad for a beginning.

The name of our acting president is Eric Liber and the name of the secretary/treasurer is Fred Kitman. We can be reached at the address below or at the following telephone numbers: Day: 412-391-3800; Night: 412-276-6546.

The members of the club have a total of 5 computers with 2 Altairs, 1 PDP/11, one Wang and one 8008. Not a bad start!! Pittsburgh Area Computer Club, 400 Smithfield St., Pittsburgh PA 15222.

Virginia Seminars

David G. Larsen of Virginia Polytechnic Institute and State University, Blacksburg VA 24061 sends in the following notice of two seminars which will be given this coming March on microcomputer themes.

1. *Microcomputer Interfacing Workshop,* March 12-13, 1976. A two-day workshop based on the popular Intel 8080 microprocessor. This course is sponsored by the VPI and SU Extension Division at the VPI Center in Reston VA (near Dulles Airport). This workshop will include many hours of experience in programming and interface construction with over 12 operating microcomputers for participant use. For more information contact Dr. Norris Bell, VPI and SU Continuing Education Center, Blacksburg VA 24061, 703-951-6328.

2. *Digital Electronics for Automation and Instrumentation,* March 21-26, which is a hands on laboratory/lecture course covering basic digital electronics as well as data communications and interfacing using asynchronous serial techniques. It is held at VPI and SU in Blacksburg VA and is sponsored by the American Chemical Society, Education Division, 1155 16th St NW, Washington DC 20036 (202)-872-4528.

The Santa Barbara California Club

Doug Penrod, 1445 La Cima Road, Santa Barbara CA 93101 sent in a good letter describing the newly formed club which has been meeting at the Goleta Library. Interested persons from the Santa Barbara area should contact Doug by mail or phone (962-3337). The club is best described by the following paragraph copied from *Datafile,* Volume 1 number 0:

The first meeting of the un-named club of computer hobbyists of the Santa Barbara area was held Wednesday evening at 7:30, 1975 October 8 at the Goleta Library, 500 North Fairview. Attendees ranged from professionals to those with a keen interest but no knowledge so far. Equipment ranged from abacus and slide rule to a Data General Eclipse system. Most expressed an interest in owning a machine of their own, from programmable calculator through microcomputer to minicomputer. Most felt that the most valuable feature of meetings is the "random access" period, during which members mill about and compare notes on their projects and problems, and examine hardware brought for display and consultation. At the first meeting was an HP-65 (Ralph Boland), a MITS Altair 8800 (Doug Penrod), a MITS Scientific Calculator (Doug Penrod), a home brew microcomputer built around an Intel 8008 microprocessor chip (Larry Plate), and a "TV Typewriter" (Grant

Runyan). Brian Johnson gave a lecture on microprocessors and computer principles for beginners, during the random access period. Everyone who is interested in any aspect of computing/calculating is encouraged to join us at our meetings. If you like mathematical games and puzzles, computer history, want to learn how computers work, want to build a computer or calculator, want to learn programming — any aspect of software or hardware, this is the place to meet the people you want to talk to. In addition to informal exchange of information, help, and goodies, we expect to have talks and demonstrations by experts on occasion. We also offer advice on publications and books.

Minneapolis-St. Paul Computer Activities

The BIT USERS ASSOCIATION is an organization of computer hobbyists and small systems engineers serving primarily the Minneapolis-St. Paul metropolitan area. The members seek to exchange newsletters and technical data with organizations or individuals with similar interests.

The BIT USERS ASSOCIATION was begun in 1973 by users of the BIT 480-483 minicomputers, and remains a source for replacement parts and licenses regarding the said machines. It has expanded to encompass persons with interest in various minicomputers and microprocessors, and is actively engaged in the design and construction of selected peripheral devices suitable for hobbyist use and which can be built at low cost at home. Raster-scan television graphic and uniquely dense alphanumeric television terminals, for example, are in final stages of development.

The sponsor of the BIT USERS ASSOCIATION is the Resource Access Center, a non-profit educational center serving community service agencies. Inquirers are requested to send self addressed stamped envelope, but no other monies please. In Minneapolis, telephone 824-6406 (or message service, 823-8247). BIT USERS ASSOCIATION, RESOURCE ACCESS CENTER, 3010 4th Ave S, Minneapolis MN 55408.

Nashua NH Area Computer Club

Dwayne Jeffries, 181 Cypress Lane, Nashua NH 03060 notes formation of a computer club in the Nashua NH area. For information and details, contact Dwayne.

Continued on page 70

Miami Area Computer Club

Terry Williamson, P.O. Box 430852, S. Miami FL 33143 went to the MITS Caravan seminar in Miami recently. As often happens when people get together with common interests, Terry got involved in conversations with several other personal computer addicts at the MITS show. The net result is the forming of a Miami area computer club. For details contact Terry by mail or at 305-271-9909. Thanks to MITS for catalyzing another computer club.

North Texas Group

The Computer Hobbyist Group of North Texas is the name of the computer club in the Dallas-Fort Worth Area. In a recent letter to BYTE, Bill Fuller (hardware editor of TCHGNT newsletter) comments:

" . . . thanks for the plugs in BYTE for the Texas Club. We went from 35 to 75 since BYTE appeared and that's why I'm so darn busy."

The club puts out a newsletter with information on activities, and has arranged for some demonstrations by computer company representatives and by members who have built computers. Meetings are held monthly. For latest details, contact Bill Fuller at 2377 Dalworth 157, Grand Prairie TX 75050.

Tiny Basic

People's Computer Company has been promoting programming in BASIC. Several companies have produced inexpensive microprocessor chips. One of these, Intel, has made the Intel 8080 chip. Currently available versions of BASIC take 4 to 8K words of memory.

PCC is working on a TINY BASIC. It will be oriented to:

kids having fun
teaching BASIC
games
elementary school arithmetic
mathematical recreations
send us your ideas . . .

It will run on an Intelec 8/Mod 80 or an ALTAIR 8800. It will use 16 bit (double word) integer arithmetic. Its design will be public so that others may reorganize the IO and mathematical subroutines for floating point. Specialized functions may be added by the user.

The proposed syntax and grammar for TINY BASIC is described in the PCC newspaper Vol. 4, nos. 1 and 2. The design philosophy is to keep it simple and use as little memory as possible. Speed is sacrificed.

PCC would greatly appreciate any help and ideas. Sixteen bit (double word) addition, subtraction, multiplication, division, decimal-binary and random number routines are needed.

Bernard R. Greening
People's Computer Company
P.O. Box 310
Menlo Park CA 94025

Washington DC Area Club Activity?

Robert Jones of Washington DC is interested in organizing a computer club for the nation's capital. If you are interested, contact him at: 4201 Massachusetts Ave, Apt. 168 W, Washington DC 20016.

Reston VA Club?

In Reston VA, Andrew Convery reports that he is interested in assembling a local club. Contact him at 2315 Freetown Ct, Apt 11C, Reston VA 22091. His home phone is (703) 860-1849, and his work phone is (703) 241-3551.

News of NECC

The New England Computer Club held its first meeting on November 5, 1975, at the Jarrell-Ashe plant in Waltham MA with about 200 people showing up. As is typical of "first meetings," the most important objective was accomplished – the assembly of a mailing list of interested individuals, and identification of a small group willing to put together a more formal organization. Future meetings have tentatively been scheduled for the Mitre Corporation cafeteria in Bedford MA.

One key goal is setting up a series of seminars on the technology of hardware and software. Other expected activities are "show and tell," trading posts for equipment exchange among individuals, and talks by manufacturers or computer users concerning applications of the technology. A newsletter will be financed by the sale of 30 Intel 8080 systems manuals donated by the local representative.

You can buy this microcomputer for $39.95, but...



MC 6800

We would be a bit surprised if you could do anything meaningful without additional hardware and software. **Wave Mate's Jupiter II**™ isn't the kind of microcomputer kit you only stare at . . . when you've completed your Jupiter II just plug in your teletype and you're ready to go. That's because it goes beyond the sum of its high quality parts. It's the ultimate micro kit experience. **In performance, in documentation, in reliability.** First, consider its superb features. It has small **pluggable wire wrapped cards** easily tailored to suit your modifications. **Every IC is socketed** and **100% burn-in tested.**

In fact every part including the powerful MC 6800 CPU and the 8K dynamic RAM is guaranteed for 120 days. It has the best software around, **System Monitor** and **Debug programs** (ROM). Includes powerful **text editor** and **Motorola compatible assembler.** And BASIC at no extra cost. Because we've been making microcomputer systems for over 4 years, we can offer you the broadest line of interfaces including TV terminal and dual audio cassette. Impressive. And yet the grandest feature is the experience of completing a kit that works. Guaranteed.

Wave Mate

ACT NOW AND SAVE.
Introductory price $1225. (Assembled $1885.)
(Good until February 1, 1976)

☐ Send Jupiter II Microcomputer Kit. Includes:

- Modular plug-in power supply
- 9 module PC backplane
- CPU module
- System monitor module
- 8K dynamic RAM module
- Serial RS 232 communication interface module
- Front panel module
- Front panel
- Wire; cut, stripped, color coded.
- Rack mount module cage
- Wire wrap tool
- Wire unwrap tool
- Cables, connectors, all other necessary hardware
- Software (editor, debug, assembler, BASIC)
- Assembly manuals
- Operators manuals
- Theory of operation manuals
- Annual membership in users group



☐ Send details on kit-a-month club
☐ Kit ☐ Assembled
☐ RS 232C conversion kit (teletype) $47.85
☐ I'm not convinced; send me 101 reasons why.
☐ Please send free Jupiter II System catalogue.

☐ Check enclosed for $ ________
Include $10.00 for postage and handling. California residents add 6% tax Delivery 60 days ARO
☐ Master Charge # ________
☐ Bank Americard # ________
Credit card expires on ________

Name ________
Address ________
City/State/Zip ________

WAVE MATE
1015 West 190th Street Gardena, California 90248
Telephone (213) 329-8941

Wave Mate

WARRANTY
120 days on all parts, materials and workmanship.

MITS Computer Caravan

The Mighty MITS Mobile stopped off in Boston recently and roused the microprocessor byters from the woodwork. It was a good deal like a club meeting as the MITS Computer Caravan paused to provide a seminar for 30 plus enthusiasts.

Since it is more than likely that virtually every one who is reading this brief report was to some degree snared into the computer hobby field as a result of the Altair 8800, we won't try to explain who MITS is or what they make.

But reading about the Altair and seeing it in action are two different things, so the crowds have been catching the Caravan in large numbers. It's a good show, so don't miss it when it comes around. The registration fee is $12 and for that price you get a loose leaf binder with schematics of the Altair and notes to go with the short instruction course presented during the Caravan seminar. The book alone makes the price reasonable. You also get an opportunity to see the Altair in action and ask questions until you get tired.

The 8800 on demo had a teletype IO and a cassette memory unit, which is what most of us were anxious to see and try out. The program was a good one, starting out with a discussion of the fundamentals of computers and logic, then the Altair hardware, and finally the software with a demo of MITS BASIC.

A couple of non-beginners went out for a short snort or something while the introduction to logic circuits unfolded . . . everyone else listened with interest. The session was run by Mike Hunter, who has been driving the Caravan all around the country.

The standard modem tones of 2025 and 2225 Hz were used for the cassette storage . . . which seemed like a good feature. After a short loading program via the front panel switches BASIC was loaded from a cassette and demonstrated. This certainly appears to be a very good version of BASIC and to have a lot of advantages over other possible languages.

The soon to be released floppy disk operating system was discussed. This will make fast access bulk memory available for a very reasonable price as compared to commercial units now available for the minicomputer systems. MITS is reportedly going to have this ready for use in a few weeks. This certainly will be a major step toward providing low cost complete operating computer systems.

The MITS line printer is also about to be released, complete with its operating system, and magnetic tape memories are in the works. Considering that all this really started only about one year ago, the progress by MITS has been just about incredible.

The MITS Caravan was thoroughly enjoyed by everyone and it seems likely that as a result of the group getting together for that seminar that a club will be forming . . . another feather in the cap for MITS. This sort of thing has been happening all around the country. ■

That Didn't Take Long at All

In the November issue of BYTE we published Dan Fylstra's article "Son of Motorola," comparing the new MOS Technology 6501 and 6502 computers with the Motorola 6800. At the time the article was written (late August) he could not predict when a kit would be available for the MOS Technology processors.

The first MOS Technology 6502 kit is now available: Microcomputer Associates Inc., of Cupertino, California, have announced the JOLT computer kit. According to their news release, this kit combines hardware and software for small scale computing at a low cost.

Components of the JOLT include the central processor kit, a 4096 byte RAM kit, an IO interface kit, power supply kit, and miscellaneous items such as a universal prototyping card and an accessory kit.

The basic central processor card contains 512 bytes of programmable random access memory, 64 bytes of interrupt vector random access memory, and 16 fully programmable IO lines. The most important feature of the CPU card is its read only memory. This memory contains the DEMON (DEbug-MONitor) program product, a basic software element needed to run the system from the serial terminal interface, which is immediately available when you turn on the power.

Classified Ads Available for Individuals and Clubs

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed or printed notice to that effect. The notices are free of charge and will be printed one time only on a space available basis. Insertions should be limited to no more than 100 words. Notices can be accepted from individuals or bona fide computer users' clubs only. Commercial advertisers should contact Bill Edwards, BYTE advertising manager, for the latest rate card and terms.

ALTAIR 8800, assembled, tested, and completely operational includes 1K RAM board with 256 words. Complete documentation including newsletters, quality built by electrical engineer. ONLY $500 postpaid. D. Schreiter, 9032 Whitehaven Dr, St. Louis MO 63123.

Would very much like to buy a used memory board (or two) for a Raytheon 704 minicomputer. Rick Loomis, PO Box 1467, Scottsdale AZ 85252, (602) 994-9104 (call afternoons or late at night).

The DEMON program includes routines to display and alter registers or memory, and control terminal interface service routines. DEMON also includes a breakpoint capability useful in debugging programs, two kinds of formatted memory read and write operations for program loading and saving, a high speed 8 bit parallel input routine and several IO service routines which may be called by a user's program.

Perhaps the most unique feature of the DEMON program is its adaptable terminal interface service routine; the software senses and automatically adapts to the speed at which the terminal operates between 10 to 30 characters per second.

A minimum system built with the JOLT components can be assembled from the CPU card kit, the accessory kit, and a Teletype or RS-232 compatible terminal. With such a system, many of the basic concepts of programming can be demonstrated within the confines of a 512 byte area of user memory. Input and output operations can take advantage of the DEMON subroutines to interact with the terminal. This would make a good initial system for a person to learn the ropes of computing.

As programming sophistication increases the owner may expand the memory by purchasing additional boards.

For users who want to breadboard their own peripherals, the universal card will come in handy as a starting point for an interface construction project.

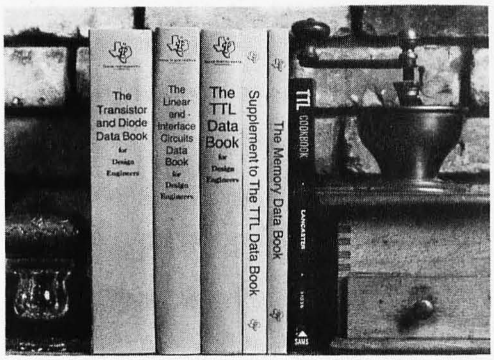
For ordering information and prices, contact Microcomputer Associates Inc., at 10440 N Tantau, Cupertino CA 95014, 408-247-8940. Delivery is 30 days after receipt of order according to their news release. ■

MICROCOMPUTER EXPERT NEEDED. I'll pay for the equipment. You put it together. We'll both use it once it's running. I'm tired of paying $1000's to a 370/155 batch processing service. My interest is statistical analysis of large volumes of stock market-type data. I need a powerful (for a micro) system, with disk or disk-type orientation, tape capabilities, line printer, card reader (maybe), BASIC or FORTRAN compiler, telecommunication ability. Joe Schachter, 41 Amory St., Cambridge MA 02139. (617) 492-2765.

USED MICROCOMPUTER EQUIPMENT NEEDED. See ad above.

ASR-33 TELETYPE in perfect physical and operating condition; 20 mA current loop interface, w/reader power supply and pedestal – $500. Also, Kleinschmidt drum printer with parallel ASCII, TTL-compatible interface, 30 cps in perfect operating condition, w/o dress cover – $300. (617) 757-2037 after 6 pm. D. F. Craft, Jr.

For sale, DEC PDP8/L mini computer. The computer is 100% working and in excellent condition. Hardware: 4K core, TTY interface, Pos I/O interface, Pos I/O bus board. Software: editor, diagnostics, assembler, other misc. Price $900. Call 617-592-2937 evenings. Peter A Balkus, Lynn MA.



You'll Want to Nybble at these Byte Books

Where does the editor of a computer magazine turn to when he must verify some author's hardware design? Information on a 75450 interface gate, or a 74147 priority encoder circuit does not spring forth by magic. Checking the information supplied by authors is part of BYTE's quality control program.

When you build a project, you need this same sort of information. All you find in the advertisements for parts are mysterious numbers identifying the little beasties . . . hardly the sort of information which can be used to design a custom logic circuit. You can find out about many of the numbers by using the information found in these books. No laboratory bench is complete without an accompanying library shelf filled with references – and this set of Texas Instruments engineering manuals plus Don Lancaster's *TTL Cookbook* will provide an excellent starting point or addition to your personal library.

● **The TTL Cookbook** by Don Lancaster, published by Howard W. Sams, Indianapolis, Indiana. Start your quest for data here with Don's tutorial explanations of what makes a TTL logic design tick. 335 pages, $8.95 postpaid.

● **The Supplement to The TTL Data Book for Design Engineers,** by Texas Instruments Incorporated. What happens when you can't find a 7400 series device listed in **The Data Book for Design Engineers?** Before you start screaming and tearing your hair out in frustration, turn to the Supplement. The Supplement has 400 pages of additional information including a comprehensive index to both TTL Data Book volumes. To be complete (and keep your hair in place and vocal cords intact) you'd best order the supplement at $1.95 to accompany the main volume.

● **The Transistor and Diode Data Book for Design Engineers,** by Texas Instruments, Incorporated. You'd expect a big fat data book and a wide line of diodes and transistors from a company which has been around from the start of semiconductors. Well, it's available in the form of this 1248 page manual from T.I. which describes the characteristics of over 800 types of transistors and over 500 types of silicon diodes. This book covers the T.I. line of low power semiconductors (1 Watt or less). You won't find every type of transistor or diode in existence here, but you'll find most of the numbers used in switching and amplifying circuits. Order your copy today, only $4.95 postpaid.

● **The Semiconductor Memory Data Book for Design Engineers,** by Texas Instruments, Incorporated. Don't forget the importance of memories to your systems. Refer to this 272 page manual to find out about the T.I. versions of many of the popular random access memories and read only memories. Order your personal copy today, only $2.95 postpaid.

● **The Linear and Interface Circuits Data Book for Design Engineers,** by Texas Instruments Incorporated. When you run across one of those weird numbers like 75365 the immediate frustration problem occurs again. What kind of gate could that be? We won't tell in this ad, but you can find out by reading the specifications in **The Linear and Interface Circuits Data Book for Design Engineers.** You can interface your brain to the 72xxx (linear) and 75xxx (interface) series of functions by ordering your copy of this 688 page manual at only $3.95 postpaid.

● **The TTL Data Book for Design Engineers,** by Texas Instruments Incorporated. How does an engineer find out about the TTL circuits? He reads the manufacturer's literature. This 640 page beauty covers the detailed specs of most of the 7400 series TTL logic devices. No experimenter working with TTL has a complete library without **The TTL Data Book for Design Engineers.** Order yours today, only $3.95 postpaid.

Caution: Heavy Reading

Buyers of these books should be cautioned: heavy reading will be required. These books are so filled with information that they weigh in at a total of about 190 ounces (5387 grams). On the basis of sheer mass, these books have got to be the bargain of the century. Make sure that you use a structurally sound book shelf and above all avoid dropping one of these books on your foot. But the mass of these books doesn't affect the bargain: we pay postage on all orders shipped to addresses in the USA and Canada, so the prices you see are the prices you pay. (That's only $.005 per gram on the average.)

____ TTL Cookbook @ $8.95
____ TTL Data Book @ $3.95
____ Supplement to TTL Data Book @ $1.95
____ Linear and Interface Circuits @ $3.95
____ Semiconductor Memory Data @ $2.95
____ Transistor and Diode Data Book @ $4.95

Send to: Name ________
Address ________
City ____ State ____ Zip ____
☐ Check enclosed
☐ Bill MC # ______ Exp. Date ____
☐ Bill BA # ______ Exp. Date ____
Signature ________

BYTE PETERBOROUGH, NH 03458

BYTE'S BUGS

Here lies documentation of known bugs detected in previous editions of BYTE . . .

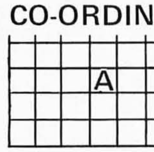
Roger Frank, 1801 E Girard #247, Englewood CO 80110 points out the following error in the text of Jim Hogenson's "Build An Oscilloscope Graphics Interface," page 72 of BYTE #2: The description of control code 7 at the top of the third column should read: "Control code 7 will not set Z, but will decrement the X coordinate of the display. This changes the horizontal section of the counter only, without affecting stored data in the memory of the interface unit."

The remaining portion of the original paragraph describing the 0 and 7 codes should be deleted, through the words "decrement X control code."

BYTE #4, page 35, table 3: The Baudot code for the letter Q in the table should read "1 1 1 0 1."

ANSWERS TO DECEMBER WORD HUNT page 18

(X, Y) CO-ORDINATES
4, 3



DIRECTION

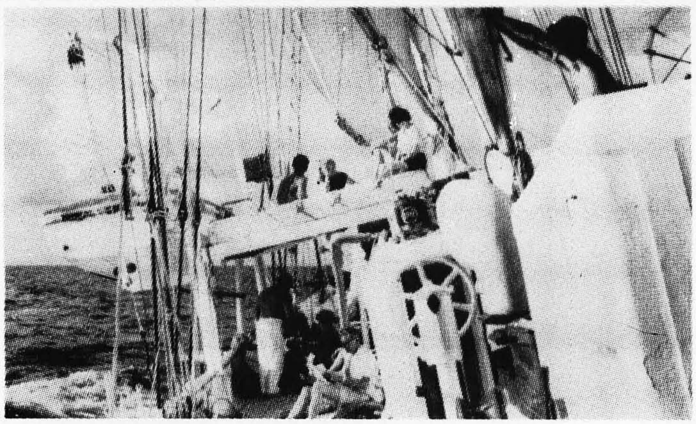


ACCUMULATOR	1, 3	8
ADD (not in address)	4, 12	2
ADDRESS	1, 15	8
ALPHANUMERIC	1, 1	2
AND	4, 3	1
APL	6, 4	5
ASCII	16, 7	6
ASSEMBLER	19, 12	7
ASYNCHRONOUS	19, 12	4
BASIC	13, 10	4
BATCH	8, 5	5
BAUD	2, 20	2
BCD	3, 10	1
BET	9, 21	2
BINARY	12, 8	4
BOOLEAN	13, 10	7
BRANCH	9, 21	4
BUFFER	14, 17	2
BYTE	9, 21	6
CHECKSUM	1, 4	1
CHIP	14, 7	8
CLOCK	9, 17	7
COBOL	13, 1	8
COUNTER	7, 13	6
DATA	6, 5	4
DEBUG	6, 5	2
DESTINATION	2, 15	2
DEVICE	1, 16	2
DIAGNOSTIC	10, 2	8
DIODE	21, 1	8
DISK	12, 18	3
DISPLAY	21, 1	6

DTL	16, 10	2
ECL	9, 2	8
EXECUTE	12, 17	6
EXIT	10, 17	8
FALSE	15, 10	4
FETCH	14, 4	8
FILE	7, 7	5
FLAG	5, 14	2
FOCAL	5, 14	6
FORMAT	5, 14	3
FORTRAN	7, 20	2
GATE	2, 12	4
GLOBAL	11, 13	2
HALT	7, 3	7
INPUT	6, 6	7
INSTRUCTION	20, 2	8
INTERFACE	20, 21	6
INTERRUPT	20, 21	4
JUMP	2, 8	1
LED	3, 17	2
LINK	2, 21	2
LIST	17, 12	8
LITERAL	13, 18	2
LOAD	2, 1	8
LOCATION	12, 13	5
LOG	14, 3	2
LOOP	4, 2	6
LSI	12, 10	6
MATRICES	8, 11	3
MEMORY	19, 20	6
MHZ	9, 6	8
MICROCODE	17, 2	6

MICROPROCESSOR	8, 11	2
MNEMONIC	17, 2	8
MOS	8, 1	8
OCTAL	6, 13	2
PCB	20, 14	5
PERIPHERAL	18, 1	8
PLOT	12, 9	8
POINTER	3, 1	8
PRIORITY	14, 19	2
PROGRAM	4, 8	1
PROM	12, 9	5
QUEUE	2, 7	1
RAM (not in program)	8, 12	2
READ	19, 17	4
RECORD	11, 8	4
ROL	14, 2	1
SOURCE	2, 18	2
STACK	16, 9	4
STATUS	11, 9	6
STORAGE	17, 14	6
SUBROUTINE	21, 14	4
SUBSYSTEM	8, 3	8
SYNTAX	21, 20	4
TABLE	12, 12	2
TRANSMITTER	10, 16	2
TRAP	11, 21	4
TTL	3, 5	2
VCC	14, 1	6
VECTOR	3, 16	4
VERIFY	3, 16	1
WIRE	4, 19	6
WORD	4, 19	5

Take off your shoes.







Hit the deck in shorts and a tee shirt. Or your bikini if you want.

You're on a leisurely cruise to remote islands. With names like Martinique, Grenada, Guadeloupe. Those are the ones you've heard of.

A big, beautiful sailing vessel glides from one breathtaking Caribbean jewel to another. And you're aboard, having the time of your life with an intimate group of lively, fun-loving people. Singles and couples, too. There's good food, "grog," and a few pleasant comforts...but there's little resemblance to a stay at a fancy hotel, and you'll be happy about that.

Spend ten days exploring paradise and getting to know congenial people. There's no other vacation like it.

Your share from $245. A new cruise is forming now. Write Cap'n Mike for your free adventure booklet in full color.

Windjammer Cruises.
A WINDJAMMER INTERNATIONAL SUBSIDIARY · OTC

Name ______________________

Address ______________________

City ______________ State ______________ Zip ______________

Phone ______________________

P.O. Box 120, Dept. 121   Miami Beach, Florida 33139

Sphere Rolls into Town

The show was a resounding success . . .

On a recent Friday evening the chaps from Sphere demonstrated their system to about 50 interested computer hobbyists at a downtown Boston Holiday Inn.

Despite an almost incredible schedule, a good show was put on. Mike Wise, the president of Sphere, was rocking a little after several days of going almost 24 hours a day . . . the caravan had driven in from Toronto that day! . . . and it was an every day a different city deal.

The product was certainly impressive . . . a computer built into a box the size of a video display and keyboard unit. That long row of switches and LEDs on most CPUs has been replaced by a little PROM software chip which permits programming to be done via the keyboard.

It was most pleasant to see the system turned on and come up ready to be used. The operating system is all built right into it in the PROM so not even a short loading program has to be switched in to get it working!

The Sphere system consists of the Motorola M6800 microprocessor with an ASCII keyboard and interface to the 6800, a video display generator system, the Motorola loader, small operating system and monitor. At $860 for the complete kit this is quite a package.

The Sphere requires an external television set for a monitor. It will feed video to this monitor, or a few parts can be added to the video output board which will generate a signal which the TV set can pick up via its antenna terminals. Some problems with FCC licensing of transmitters seems to have held up the supplying of these parts with the kit itself.

Mike told the group that they are well along with a BASIC language package for the system and that they expect to have a library of about 4000 business oriented programs available.

The Sphere literature stresses the business applications of the system, showing it in use in offices, stores, homes, banks, laboratories, etc. Since this is the major potential for small computer systems this is certainly an enlightened approach. As more and more computer stores open up around the country and literature such as that provided by Sphere is gotten out to the owners of small businesses, sales of these systems cannot be but astronomical.

The Sphere sales literature is extremely well done . . . and best of all, it is written for the businessman, not the computer expert. Computer stores should have little trouble in selling this approach to the business world. The utter simplicity of the system will certainly appeal to the businessman.

The PROM has quite a repertoire and includes an editor, assembler, debugger, command language, cassette loader, dumper and utilities . . . all on a 1 KB chip! Since the programming is on a PROM, this still leaves the entire 4 KB RAM available for the user and doesn't tie 1 KB of it up just with the operating system . . . a big plus even at today's low memory prices.

The PROM operating system permits rather sophisticated editing techniques including movement of the cursor with ease, scrolling of the lines of text through the buffer memory, delete lines, add lines, modify lines, etc. The system also permits the keyboard to be used for entering programs into the computer using hexadecimal or octal notation. It also enables the user to look at any address and see what is there . . . change the data if desired . . . and then scan on through succeeding bytes. Not a bad bunch of programs to be crammed into one 1024 byte PROM!

Another plus is the comprehensive Sphere book of instructions. It is not only amazingly complete (particularly when you consider how short a time the company has been in existance), it is also written in language which is not beyond the relative newcomer to computers.

In all, the show was a resounding success in Boston and a tired, but happy Sphere crew packed up for the next night in New York. ■

THE SUNTRONIX MODEL KBD IV Keyboard is ideally suited as a general purpose ASCII Keyboard for data terminal applications. This keyboard more than meets the needs of the data entry market for long life and reliability.

The KBD IV utilizes the 2 key rollover solid state read-only MOS memory allowing encoded outputs to be strobed out as each key is depressed. A second key may be depressed concurrent with the first, but the second key encoded output will not be strobed out until the first key is released. This feature prevents ambiguity of character codes as a result of two keys being depressed in rapid succession.

ELECTRICAL SPECIFICATIONS:

* Voltage requirements — +5.0 V and −12.0 V
* Power consumption — less than 200 mw
* Outputs — standard ASCII; 7 bits + strobe
* Negative or positive logic output, jumper selectable
* Output connector — standard 14 pin DIP IC socket
* Three modes — normal, shift & control

MECHANICAL FEATURES:

* Size — 12¼" x 6¾" x 2½"
* High grade glass epoxy PC board
* Keyboard ROM SMC KR2376 40 pin MOS
* Electronic shift lock, not mechanical
* Keyswitches one integral assembly, not individual keys
* Switches have four-finger phosphor bronze contacts with gold inlay
* Keycaps are 2-shot high strength ABS plastic

These keyboards are available off the shelf in two forms — fully assembled and unconditionally warranteed against defects in manufacture or materials for a full ninety days, or in the more economical kit version. Kit parts are fully warranteed against defects in manufacture or materials for a full ninety days. In either version, full instructions are supplied for operation, including specifications and data sheets. In the kit version, complete instructions are supplied for the assembly process. A reasonably competent technician can completely assemble and test this keyboard in one evening. All parts needed are included.

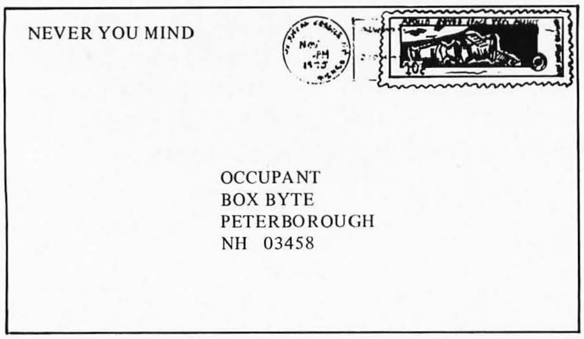
INTRODUCTORY PRICES

Factory assembled — $74.95 ppd.
Complete kit, w/instructions — $64.95 ppd.
Please add $1.00 handling per order.
Minimum order — $5.00



SUNTRONIX company

360 Merrimack St., Lawrence, Mass. 01843
(Telephone orders: 617-688-0751 603-434-4644)



TASTY BUT NOT FILLING

A friend of mine showed me the first three issues of your new magazine. I found them extremely tasty but not at all filling; please send more BYTEs as soon as possible! (I have sent my subscription order in.)

I have just one "bit" of criticism: Please adopt some designations for end-of-article, continued-on-next-page, and continued-on-page-n! Their use is not *essential*, but my ETP (English Text Processor) has become used to finding them (it's a heuristic routine, you see) and often wastes considerable time when they're not there.

Possibilities include the IBM end-of-file code, /*, for end-of-article, or the ASCII abbreviations ETX (end of text) and FF (form feed) for end-of-article and go-to-next-page, respectively. And FF nn could be used for go-to-page-nn. Alternatively, some common notations from assembly languages could be used: END, BR *+1 (go to next page), and BR nn (go to page nn); but on second thought, these would probably not be as easily recognized by as many people as the ASCII symbols.

Hope to see your next issue in my mailbox. Meanwhile, I'm going to try to persuade the proprietors of my college bookstore to carry BYTE. Wish me luck!

Jamie E. Hanrahan
Fullerton CA

Regarding end of file marks, you should notice by now that articles in BYTE's latest issues are terminated by a typeset "■". We've considered other things ranging from an "EOF" to an END statement. This convention of the black square is simple and effective, so it became the obvious choice.

Once that happy time comes to the ham, linking home brew computer terminals to their Teletype set-ups will combine two of the most fascinating hobbies ever conceived by the minds of men.

WORDS FROM A READER

Your answer to a reader's question on the difference between a "bit" and a "byte" and a "word" was pretty good, except for the definition of a word. The difference is made more apparent if we look at the evolution of computers.

Early, so-called first generation machines generally performed arithmetic operations only in accumulators, which were of a fixed bit length. This length was called a word. Since all operations were done in accumulators, each storage address usually had the same bit length, so a word was also the number of bits obtained in one storage reference.

In the second generation, systems were classified as either business or scientific. The scientific machines generally used binary arithmetic, performed in accumulators as before. Business systems, however, did not have accumulators, but used storage-to-storage arithmetic. In this system, the contents of two storage locations were input to the ALU, and the ALU output was stored in one of those locations. These machines were of variable word length, in that a non-data bit in each storage location was used to mark the end of a word. Since most of these machines used 6 data bits and each of the 64 bit combinations had an assigned graphic, each storage location was usually referred to as a character.

The IBM 360 series was the first of the third generation, and was designed to be general purpose; that is, either scientific (binary) or business (decimal). Its general purpose registers (which could be used as binary accumulators or index registers or memory address registers) were 32 bits long, so this was called a word. The floating point accumulators were 64 bits, or a double word. Thus the binary arithmetic instructions could obtain, in one storage reference, either a double word, a word, or a half word (16 bits). Decimal arithmetic was performed storage-to-storage, with the instruction containing the length of each operand. Since the term word was used to refer to the register (accumulator) length, a new term was needed to refer to the number of bits contained at any single storage address; IBM chose the term byte, and in the 360 series a byte was 8 bits.

In most micros to date, the number of bits at one storage address (a byte) is the same as the length of the accumulator (word), so the terms are interchangable, whether it is 8, 12, or 16 bits. But who knows what the future will bring?

R. S. Downs
Raleigh NC

OSCILLOSCOPE GRAPHICS INTERFACE

The Oscilloscope Graphics Interface in the October 1975 BYTE is a useful addition to any microcomputer system. However, if the display ever got really busy, it looks like the screen would be a mess. This is due to the fact that when a Set X or Set Y is used, the scan continues from that point on down the screen. Rapid access to the 2102's, as in any sort of "moving picture" presentation, would never let the trace get down the screen very far.

A simple modification which would alleviate the situation would be to have two counters feeding the 2102 address lines, one free running at 100 kHz all the time, and the other under program control. With three state buffers (i.e., 8097's), the memory write of new information would gate only the desired address to the 2102's, after which the other address (from the 100 kHz counters) would again be applied to the 2102's. Thus the scan would pick up shortly after the point where it left off.

For readers with an infrequent screen update, this is not necessary, but for those with fast processors doing frequent display modification, this approach should work nicely.

Roger Frank
Englewood CO

You are right, Roger, and the hardware modification you propose will correct the situation. Another way to avoid running into that problem is to use appropriate software: Simply, do full screen updates when changes are desired.

THOUGHTS ON TELETYPE

You have asked for comments from BYTE subscribers about themselves, and anything else that may come to mind, and here are my thoughts on what I had hoped for and anticipate in your magazine.

First off, I am a ham, W2ZPW, for some 26 years. That automatically signifies a General class ticket or better, and General class it is. I am not interested in any higher class of license for a very good reason. I am entirely deaf. Any additional frequencies earned with a higher class of license would require *hearing* for phone or CW (*Continuous Wave* or Morse code for anyone who just immigrated from the Moon).

My sole interest in ham radio is in Teletype. Since computer technology runs closely parallel, in many ways, with radio

frequency communications, there is no doubt in my mind but that a goodly proportion of articles to come will be slanted toward the amateur radio enthusiast. Considering this possibility it is certainly not too much to hope for to ultimately see a simple and inexpensive commercial ASCII-to-Baudot converter that could readily be used to actuate the widely used Baudot coded amateur Teletype equipment.

Use of ASCII code is being pondered and will no doubt soon be written into FCC regulations. But even so, those amateurs who have RTTY using five level Baudot equipment will mostly continue to do so because it is in virtual universal use. It is my considered opinion that ASCII will take hold to a greater degree than 100 WPM Baudot did, but by means of soon to be devised converters (of a simple and inexpensive nature, it is hoped), switching from Baudot to ASCII, and back again, will become virtually universal.

Once that happy time comes to the ham, linking home brew computer terminals to their Teletype set-ups will combine two of the most fascinating hobbies ever conceived by the minds of men (and women). It then becomes possible to visualize a means of communication combining cable and radio. Where more than one mini-computer system is linked by way of telephone lines it will become a simple matter to "patch" the phone line into a radio station and rebroadcast the computer output to form input to another computer repatched back into a phone system elsewhere at a distant point. It opens entirely new vistas that may take a bit of deep thinking to even visualize at this point. The possibilities are so broad in scope that I hate to even begin to enumerate them.

In view of the FCC restriction made upon hams to use only five level Baudot code, most of the amateur equipment in current use is American made Teletype Corporation equipment. Some model 14, mostly tape gear, is used. Model 15 printers and the companion model 19 sets combine model 14 and 15 to form a composite set that will either produce coded tape for transmission (but not be received from on the air) or print out on rolls copy that is received or sent. The model 15 printers are often obtainable for as low as $25 and rarely higher than $100 for the model 19.

The ultimate in Baudot equipment at present is the model 28. Obsolete as they may be where computers are concerned, they are highly prized for ham use. There are few model 33 and 35 Teletype machines amongst amateurs but since they are designed for ASCII use they require some means of conversion to Baudot coding.

As soon as the FCC removes the restriction against ASCII coding, a magazine such as BYTE will then become the obvious vehicle for guiding both amateur radio and computer enthusiasts to combine their hobbies for the greatest enjoyment imaginable. The possibilities are endless and limited only by the imagination.

Mack O. Santer
Brooklyn NY

It looks like there is plenty of possibility for the synthesis of amateur radio and computer activities. It will be interesting to see just what people actually do with this novel application of computers. BYTE will report on developments in the radio/computer field as they occur.

QUESTIONS – STAR TREK, SURPLUS CHIPS

In the first three issues of BYTE, which are just what the doctor ordered, you've mentioned the game of Star Trek a number of times in your columns. I am a Star Trek freak with an IBM 1130 at my disposal. Is there any way of obtaining a Star Trek program written in Fortran IV or other information on the subject?

In the near future, will any companies be marketing a 16 bit computer utilizing National's IMP-16 CPU chip?

What companies carry used or surplus Teletype machines in working condition at reasonable prices? Perhaps your magazine can do a series on the surplus market place.

In the first issue you published a review of the RGS 008A computer by James Hogenson. I found this very informative and am looking forward to more such reviews, perhaps on the Altair 8800 or SWTPC's new Motorola based 6800 computer system.

Looking forward to more fantastic issues of BYTE.

Richard Wexler
Texarkana TX

There is a BASIC version of Star Trek called Space War listed in complete detail in 101 BASIC Games, *published by DEC in Maynard MA. See the book review in this issue. The closest product to an IMP-16 yet available is the National PACE chip, its LSI single-chip successor.*

Continued on page 86

5% OFF ON ORDERS OVER $50.00
10% OFF ON ORDERS OVER $100.00
15% OFF ON ORDERS OVER $250.00

JANUARY SPECIALS

DIGITAL

7447	BCD to 7 seg. decoder drvr.	$ .69
7483	4 bit binary full adder	.59
7490	Decade counter	.45
74151	8 channel multiplexer	.59
74153	Dual 4 input multiplexer	.79
74180	Parity generator checker	.79
74193	Binary up/down counter	.99
9602	Dual one shot.69	
75C00	Quad 2 input NAND gate (CMOS)	.12

LINEAR

301	Hi perf op amp	mDIP	.25
339	Quad comparator	DIP	1.29
320T	5V neg. V reg.	5V	1.39
320T	12V neg. V reg.	12V	1.39
340T	12V neg. V reg.	5V	1.39
340T	15V pos. V reg.	15V	1.39

MEMORIES

2102	$2.65
2102-1	2.75
2102-2	2.75
82S23	3.25
F93410	
256 bit RAM	2.19

CLOCK CHIPS

5311	$3.95
5312	2.95
5314	3.95
5316	4.15

CALCULATOR CHIPS

5005	$1.69

8038 FUNCTION GENERATOR
Voltage controlled oscillator — sine, square, triangular output. 16 pin DIP with data **$3.95**

7001 CLOCK CHIP
4-6 digit, 12-24 hr. alarm, timer and date circuits — with data **$6.95**

DVM CHIP 4½ DIGIT
MM5330 — P channel device provides all logic for 4½ digit volt meter. 16 pin DIP with data **$14.95**

TTL

		7451	.17	74154	1.25
7400	$ .14	7453	.17	74155	1.07
7401	.16	7454	.17	74156	1.07
7402	.15	7460	.17	74157	.99
7403	.16	7464	.35	74158	1.79
7404	.19	7465	.35	74160	1.39
7405	.19	7470	.30	74161	1.25
7406	.35	7472	.30	74162	1.49
7407	.35	7473	.35	74163	1.39
7408	.18	7474	.35	74164	1.59
7409	.19	7475	.57	74165	1.59
7410	.16	7476	.39	74166	1.49
7411	.25	7483	.79	74170	2.30
7413	.55	7485	1.10	74173	1.49
7416	.35	7486	.40	74174	1.62
7417	.35	7489	2.48	74175	1.39
7420	.16	7490	.59	74176	.89
7422	.26	7491	.97	74177	.84
7423	.29	7492	.71	74180	.90
7425	.27	7493	.60	74181	2.98
7426	.26	7494	.94	74182	.79
7427	.29	7495	.79	74184	2.29
7430	.20	7496	.79	74185	2.29
7432	.23	74100	1.30	74187	5.95
7437	.35	74105	.44	74190	1.35
7438	.35	74107	.40	74191	1.35
7440	.17	74121	.42	74192	1.25
7441	.98	74122	.45	74193	1.19
7442	.77	74123	.85	74194	1.25
7443	.87	74125	.54	74195	.89
7444	.87	74126	.63	74196	1.25
7445	.89	74141	1.04	74197	.89
7446	.93	74145	1.04	74198	1.79
7447	.89	74150	.97	74199	1.79
7448	1.04	74151	.79	74200	5.90
7450	.17	74153	.99		

LOW POWER TTL

74L00	$ .25	74L51	$ .29	75L90	$1.49
74L02	.25	74L55	.33	74L91	1.45
74L03	.25	74L71	.25	74L93	1.69
74L04	.25	74L72	.39	74L95	1.69
74L06	.25	74L73	.49	74L98	2.79
74L10	.25	74L74	.49	74L164	2.79
74L20	.33	74L78	.79	74L165	2.79
74L30	.33	74L85	1.25		
74L42	1.49	74L86	.69		

HIGH SPEED TTL

74H00	$ .25	74H21	$ .25	74H55	$ .25
74H01	.25	74H22	.25	74H60	.25
74H04	.25	74H30	.25	74H61	.25
74H08	.25	74H40	.25	74H62	.25
74H10	.25	74H50	.25	74H72	.39
74H11	.25	74H52	.25	74H74	.39
74H20	.25	74H53	.25	74H76	.49

8000 SERIES

8091	$ .53	8214	$1.49	8811	$ .59
8092	.53	8220	1.49	8812	.89
8095	1.25	8230	2.19	8822	2.19
8121	.80	8520	1.16	8830	2.19
8123	1.43	8551	1.39	8831	2.19
8130	1.97	8552	2.19	8836	.25
8200	2.33	8554	2.19	8880	1.19
8210	2.79	8810	.69	8263	5.79
				8267	2.59

9000 SERIES

9002	$ .35	9309	$ .79	9601	$ .89
9301	1.03	9312	.79	9602	.79

CMOS

		4016A	.56	4050A	.59
4000A	$ .26	4017A	1.19	4066A	.89
4001A	.25	4020A	1.49	4068A	.44
4002A	.25	4021A	1.39	4069A	.44
4006A	1.35	4022A	1.10	4071A	.26
4007A	.26	4023A	.25	4072A	.35
4008A	1.79	4024A	.89	4073A	.39
4009A	.57	4025A	.25	4075A	.39
4010A	.54	4027A	.59	4078A	.39
4011A	.29	4028A	.98	4081A	.26
4012A	.25	4030A	.44	4082A	.35
4013A	.45	4035A	1.27	4528A	1.60
4014A	1.49	4042A	1.47	4585A	2.10
4015A	1.49	4049A	.59		

74C00	$ .22	74C74	$1.04	74C162	$2.93
74C02	.26	74C76	1.34	74C163	2.66
74C04	.44	74C107	1.13	74C164	2.66
74C08	.68	74C151	2.61	74C173	2.61
74C10	.35	74C154	3.15	74C195	2.66
74C20	.35	74C157	1.76	80C95	1.35
74C42	1.61	74C160	2.48	80C97	1.13
74C73	1.04	74C161	2.93		

PULTRON V.I.P. QUARTZ REGULATED LED WATCH

MEN's 5 function — hours, minutes, month, day and seconds.
Latest slim-line design (prox. ⅜" thick)
Four year calendar — requires setting only on February 29th.
One year warranty against all defects in material and workmanship.
Built-in phototransistor adjusts intensity of display.
Gold tone — with adjustable mesh strap.



$99.50
+$1.50 Shipping & Handling
Usual 10% Discount Applies

Data sheets on request. With order add $.30 for items less than $1.00 ea.

LINEAR CIRCUITS

300	Pos V Reg (super 723) TO-5	$ .71
301	Hi Perf Op Amp mDIP TO-5	.29
302	Volt follower TO-5	.53
304	Neg V Reg TO-5	.80
305	Pos V Reg TO-5	.71
307	Op AMP (super 741) mDIP TO-5	.26
308	Mciro Pwr Op Amp mDIP TO-5	.89
309K	5V 1A regulator TO-3	1.35
310	V Follower Op Amp mDIP	1.07
311	Hi perf V Comp mDIP TO-5	.95
319	Hi Speed Dual Comp DIP	1.13
320	Neg Reg 5.2, 12, 15 TO-3	1.19
322	Precision Timer DIP	1.70
324	Quad Op Amp DIP	1.52
339	Quad Comparator DIP	1.58
340K	Pos V reg (5V, 6V, 8V, 12V, 15V, 18V, 24V) TO-3	1.69
340T	Pos V reg (5V, 6V, 8V, 12V, 15V, 18V, 24V) TO-220	1.49
372	AF-IF Strip detector DIP	2.93
373	AM/FM/SSB Strip DIP	.53
376	Pos V Reg mDIP	2.42
377	2w Stereo amp DIP	1.16
380	2w Audio Amp DIP	1.13
380-8	.6w Audio Amp mDIP	1.52
381	Lo Noise Dual preamp DIP	1.52
382	Lo Noise Dual preamp DIP	.71
550	Prec V Reg DIP	.89
555	Timer mDIP	.89
556A	Dual 555 Timer DIP	1.49
560	Phase Locked Loop DIP	2.48
562	Phase Locked Loop DIP	2.48
565	Phase Locked Loop DIP TO-5	2.38
566	Function Gen mDIP TO-5	2.25
567	Tone Decoder mDIP	2.66
709	Operational AMP TO-5 or DIP	.26
710	Hi Speed Volt Comp DIP	.35
711	Dual Difference Compar DIP	.26
723	V Reg DIP	.62
739	Dual Hi Perf Op Amp DIP	1.07
741	Comp Op Amp mDIP TO-5	.32
747	741 Dual Op Amp DIP or TO-5	.71
748	Freq Adj 741 mDIP	.35
1304	FM Mulpx Stereo Demod DIP	1.07
1307	FM Mulpx Stereo Demod DIP	.74
1458	Dual Comp Op Amp mDIP	.62
1800	Stereo multiplexer DIP	2.48
LH2111	Dual LM 211 V Comp DIP	1.70
3900	Quad Amplifier DIP	.35
7524	Core Mem Sense AMPL DIP	.71
8038	Voltage contr. osc. DIP	4.25
8864	9 DIG Led Cath Drvr DIP	2.25
75150	Dual Line Driver DIP	1.75
75451	Dual Perepheral Driver mDIP	.35
75452	Dual Peripheral Driver mDIP	.35
75453	(351) Dual Periph Driver mDIP	.35
75491	Quad Seq Driver for LED DIP	.71
75492	Hex Digit driver DIP	.80

LED's

MV10B	Red TO 18	$ .22
MV50	Axial leads	.18
MV5020	Jumbo Vis. Red (Red Dome)	.22
	Jumbo Vis. Red (Clear Dome)	.22
ME4	Infra red diff. dome	.54
MAN1	Red 7 seg. .270"	2.19
MAN2	Red alpha num .32"	4.39
MAN4	Red 7 seg. .190"	1.95
MAN5	Green 7 seg. .270"	3.45
MAN6	.6" high solid seg.	4.25
MAN7	Red 7 seg. .270"	1.19
MAN3	Red 7 seg. .127" straight pins	.29
MAN8	Yellow 7 seg. .270"	3.45
MAN66	.6" high spaced seg.	3.75
MCT2	Opto-iso transistor	.61

MEMORIES

1101	256 bit RAM MOS	$ 1.50
1103	1024 bit RAM MOS	3.95
1702A	2048 bit static PROM UV eras.	17.95
2102-2	1024 bit static RAM	4.25
5203	2048 bit UV eras PROM	17.95
5260	1024 bit RAM	2.49
5261	1024 bit RAM	2.69
5262	2048 bit RAM	5.95
7489	64 bit ROM TTL	2.48
8223	Programmable ROM	3.69
74200	256 bit RAM tri-state	5.90

MULTIPLE DISPLAYS

NSN33	3 digit .12" red led 12 pin fits IC skt.	$1.79
HP45082 7405	5 digit .11 led magn. lens com. cath	3.49
HP5082 7414	4 digit .11 LED magn. lens comm. cath.	3.25
FNA37	9 digit 7 seg led RH dec clr. magn. lens	4.95
SP-425-09	9 digit .25" neon direct interface with MOS/LSI, 180 VDC, 7 seg.	1.79

CALCULATOR & CLOCK CHIPS

5001	12 DIG 4 funct fix dec	$2.49
5002	Same as 5001 exc btry power	2.79
5005	12 DIG 4 fuct w/mem	2.99
MM5725	8 DIG 4 funct chain & dec	1.98
MM5736	18 pin 6 DIG 4 funct	4.45
MM5738	8 DIG 5 funct K & mem	5.35
MM5739	9 DIG 4 funct (btry sur)	5.35
MM5311	28 pin BCD 6 dig mux	4.45
MM5312	24 pin 1 pps BCD 4 dig mux	3.95
MM5313	28 pin 1 pps BCD 6 dig mux	4.45
MM5314	24 pin 6 dig mux	4.45
MM5316	40 pin alarm 4 dig	5.39

SHIFT REGISTERS

MM5013	1024 bit accum. dynamic mDIP	$1.75
MM5016	500/512 bit dynamic mDIP	1.59
SL5-4025	QUAD 25 bit	1.29

DTL

930	$ .15	937	.15	949	.15
932	.15	944	.15	962	.15
936	.15	946	.15	963	.15

Satisfaction guaranteed. Shipment will be made via first class mail in U.S., Canada and Mexico within 5 days from receipt of order. Add $.50 to cover shipping and handling for orders under $25.00. Minimum order $5.00. California residents add sales tax.

INTERNATIONAL ELECTRONICS UNLIMITED
P.O. BOX 1708/ MONTEREY, CA. 93940 USA
PHONE (408) 659-3171

NYBBLE

At long last one of the things that I have wondered about all my life has been defined. I refer to your answer to Stephen Holland's letter on page 84 of issue 3. You define a nybble as half of a byte. I am a fisherman and had never before known exactly what a nybble was. Now will you define stryke?

All kidding aside, I enjoy the magazine greatly. I am a nuts and bolts hardware type with a good technical background, first phone and amateur extra, and enjoy all the articles. The ones on programming are above me yet. No experience in that area. So, on page 88 I give them all 10 (shades of *Analog*).

Keep up the good work with a balanced magazine that will appeal to all of us for we all need to know something about each other's areas.

Roy W. Dancy
Dothan AL

A stryke is what happens when lightning hits a computer. It gets all shaken up and refuses to work.

ROLLING HIS OWN

I am particularly encouraged to see a magazine of this type published; most of the other publications available either assume the background of an electrical engineer or computer science theoretician, or assume no background at all. I've always wanted a computer of my own, and when the "Mark-8" project came along, I jumped at the chance to get one. I have it almost finished, but have had some problems in obtaining simple items like a decent power supply at a reasonable price (not trusting my electrical theory enough to build my own), or attractive paddle switches for a front panel. Fortunately I have not been working alone, having a partner building another Mark-8 along with me; we order our components together and take advantage of quantity discounts.

Your magazine kept my wheels turning into the early hours of the morning, especially the material on graphics display devices. The 8008 is cheap and that is an important consideration for anyone on a grad student's budget. Perhaps in the future, the other chips and kits will come down in price and us poor folks will get a chance to play with the other stuff.

Robert Jones
Washington DC

PS, Any chances of an organized project to get a hardware text editor, loader, assembler for 8008 systems?

It is good to hear that you are getting a computer up and running, Robert. An 8008 will certainly serve you faithfully and reliably once you get your system up, running and debugged. I may be critical of the 8008 at times because it is a machine with which I have intimate familiarity: my first home brew computer learning experience was an 8008. Aside from the relatively slow speed of an 8008, it is quite capable of executing all the typical home brew computer person's requirements.

A well done Spacewar game is one of the most interesting strategy and hand-eye coordination tests ever invented.

SPACEWAR, ANYONE?

Let me first add a word of appreciation to all the others you have received. BYTE is a great magazine. Be sure and send me my renewal notice when it comes due.

The other item I would like to bring up is Spacewar. This is a game that was played on conventional computers 5 years ago. How much memory does it take? Do you know where I could get a copy of a program of one of its versions?

Stewart Shelton
Madison WV

Thank you Stewart. We cannot reel off a definite answer on memory requirements for Spacewar, but do believe a reasonably competent Spacewar application is within the realm of the home brew computer technology as it stands today. We've seen implementations done on minicomputers with less than 32K 16-bit words. When we get some good articles on this application, you'll see them printed in the pages of BYTE. A well done Spacewar game is one of the most interesting two-person strategy and hand-eye coordination tests ever invented.

HOUSEKEEPING COMPUTER

My husband just received his first copy, October, of BYTE yesterday, and as usual I glanced at it as I brought it in. Surprise! An article just for me! Most of his magazines don't interest me too much, because their articles are too technical. But Richard Gardner's article was fascinating. We almost fought over who got to read the magazine.

The main reason the article was so interesting to me is that that is almost exactly our story. We have our own computer (Altair 8800 with 9K of memory), and we have been planning to use it in most of the ways you mentioned. Soon I will be allowed to put my food planning on the computer: recipes, menus, nutritional planning, seasonal shopping, etc. I think I'll even key it to alternate speedy meals for days when dinner starts late. Keith has already begun thinking of eventually using it to control our interior environment, which will be very helpful in maintaining proper storage conditions for our year's food supply. I think he is even planning to use it to play selected music throughout our dream house from a central control center to speakers in the various rooms. His original plan was to develop an information retrieval system, to reduce the space required in filing. (We have a 4-drawer legal size file cabinet, full, as well as many books, notebooks, etc.) And we have already been having the neighbor kids in playing with the computer. When we get the terminal and cassette recorder connected, our only limit will be the sky (or is it the memory?).

I'm looking forward to hearing more about home computers, and would appreciate help in programming my food programs, as I don't know much of the technical aspects.

Carol H. Kendall
Vernal UT

LOGICAL INERTIA

Yes indeed, by all means, instantly enter my subscription to your fabulous, fascinating, incredible magazine. My imagination races at the thought of a Klugeharp. TV typewriters titilate my thoughts. Do it . . . my check is enclosed.

Allow me to share with you a notice posted on the wall of a major computer manufacturer I once was associated with:

> Today's computers execute instructions in the sub-microsecond range. This can cause programs to develop considerable logical inertia, making HALTS extremely dangerous. The following technique is suggested when a HALT is necessary:

```
Stop  HALT
      HALT   In case still skidding
      GOTO   Stop
```

L. E. Staples
San Antonio TX

As with all software problems, the solution is a mass *of instructions.*



BOOK REVIEWS

101 BASIC Computer Games, *Digital Equipment Corporation. Maynard, Mass. $7.50.*

Because they are simply programmed, there are endless small games of chance, games involving out-guessing the computer, and venerable "ancient" games, often with modern twists thrown in.

Before you dismiss this as just another of the plethora of BASIC game books, note that *101 BASIC Computer Games* claims the distinction of being the *"first* [circa July 1973] collection of games all in BASIC," and "the only collection that contains both a complete listing *and* a sample run of each game along with a descriptive write-up."

The preface cites the educational value of games as one of the prime motivations for the book. This is a fact which no one would seriously contest, but which computer center directors seem to overlook as they contemplate the hours of wasted computer time. Since time on our home systems is never wasted, by definition, this shouldn't pose a problem.

Most of the games in the book are computer simulations of simple games of chance, sports, "ancient" games, and board or card games. And, as the preface adds, very few of the games use the innovative capabilities of the computer "to come up with something new and truly unique."

The authors of this collection have, for the most part, put an imaginative or humorous illustration with each game. The format consists of a short descriptive write-up of the game, any special language or computer requirements, a reference to (and sometimes personal notes from) the author, and a listing of the game, along with sample runs.

Many of these programs are over-simplified versions of the "real" games, due to the dual constraint of the BASIC language and generally limited computer resources, but this makes them even more attractive for the microcomputer user.

Because they are simply programmed, there are endless small games of chance, games involving out-guessing the computer, and venerable "ancient" games, often with modern twists thrown in. For example, there's AWARI, an ancient game of rotating beans in a pit; BAGLES, a 3-digit number guessing game that teaches simple logic; BUG, a dice-rolling game that draws a bug part-by-part; CHOMP, involving eating a computerized cookie, while trying to avoid eating the poisoned piece; the ancient game of GUESS, where the user tries to guess the number the computer is "thinking of"; HANG, the age-old game of hang-man, with the added "bell and whistle" of drawing the hung man part-by-part; and TOWER, a game program that helps the user attempt to solve the old, old puzzle of the Towers of Hanoi, providing a graphic display of each step on the way.

There are dozens of standard game-playing or sport-simulating programs. For example, in the card game area, we have ACEYDU, an acey-deucey game; BLKJAC, a very comprehensive (Las Vegas rules) black-jack game; POKER; and the old card game of WAR. The book also sports baseball, basketball, bowling, boxing, darts, bull-fighting, high school and professional foot-ball, golf, ice-hockey (against Cornell!), and parachuting games. Board games include bingo, checkers, Go, Monopoly, Nim, and three-dimensional Tic-Tac-Toe (none of which embody much computer intelligence, unfortunately). The simulations include games of battleship, gunboat, a World War II bombing mission, Can-Am road racing, Civil War battles, fur-trading, governing the ancient land of Sumeria, ruling a modern island wisely, successful pizza-delivering, Apollo capsule landing, lunar lander maneuvering, space war, and a stock market simulation. The gambling games and games of chance are even more numerous.

Many of these programs are over-simplified versions of the "real" games, due to the dual constraint of the BASIC language and generally limited computer resources, but this makes them even more attractive for the microcomputer user. A few of the games *are* quite comprehensive and thus rather large: blackjack, pro football, Can-Am racing, lunar landing, and space war (which is really a complete Star Trek simulation).

101 BASIC Computer Games includes a few novelty and scientific programs such as BANNER, which prints large letters on standard terminals in a banner-like fashion,

S. D. SALES CO.

6 Digit Digital Clock Kit

We wanted **our word** to mean something to **you**. When we first introduced our clock we used such words as "impossible", "unbelievable bargain", "prime quality", and "unconditional money back guarantee." We anticipated selling several hundred kits. Your response to our clock has been fantastic. Now several thousand kits later, we hear such comments as "outstanding value", "exceptional service", "I can't believe it", etc. One enthusiastic customer even wrote saying it was "the best thing since bottled beer."

Here's What The Kit Includes:

1 - MM5314 National Clock Chip with socket
6 - Common Cathode Led Readouts (.25 in. char.)
13 - NPN and PNP Driver Transistors
2 - Push Button Switches for time set
1 - Switch for Time Hold
1 - Filter Cap
4 - IN4001 Rectifiers
1 - IN914 Diode
2 - .01 Disc. Caps
9 - Carbon Resistors

$9.95 (KIT) WITH SPECS. POSTPAID

Transformer $1.50

P.C. BOARD FOR ABOVE $3.00 each

INTEL 1702A 2K ERASEABLE PROM'S $6.95

We tell it like it is. We could have said these were factory new, but here is the straight scoop. We bought a load of new computer gear that contained a quantity of 1702A's in sockets. We carefully removed the parts, verified their quality, and are offering them on one heck of a deal. First come, first served. Satisfaction guaranteed.

PROFESSIONAL QUALITY STEREO HEADPHONES

Here is the kind of super deal that S. D. is famous for. Treat your ears to a super sound at a super price. Soft padded ear cushions, lightweight, fully adjustable headband, long coiled cord, wide response. **$6 Post Paid - NEW IN ORIGINAL FACTORY BOXES**

SALE ON CUT LEAD SEMICONDUCTORS

Leads were cut for PCB insertion. Still very useable. IN4148 Diode - 40/$1 2N3904 (House #Transistor) - 25/$1

INSTRUMENT KNOBS

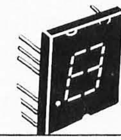
Black with brushed aluminum insert. Medium size, very attractive style. **SPECIAL 5 FOR $1**

VERNIER DIAL

From a close out of metal detector manufacturer. ½ Turn, 8 to 1 ratio. Internal stops easily removed to make unit multi-turn.

LIMITED QUANTITY - $.99 EACH

DO YOU NEED A LARGE COMMON ANODE READOUT AT A FANTASTIC PRICE !

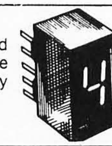


S. D. presents the MAN-64 by Monsanto - .40 inch character. All LED construction - not reflective bar type, fits 14 pin DIP. Brand new and factory prime. Left D.P.

$1.59 ea. 6 for $7.50

FAIRCHILD LED READOUTS - 69c

FND-70. Common cathode. Right hand decimal point. .25 inch character. Prime new units. Perfect for clocks, frequency counters, etc. 69c 10 For $6
Best Readout Buy in USA!



DRIVER TRANSISTORS
2N3904 - NPN
2N3906 - PNP
8 for $1

2102 1K RAM's - 8 FOR $12.95

New units by National. We bought a load on a super deal, hence this fantastic price.
Units tested for 500NS Speed.

FET'S BY TEXAS INSTRUMENTS

#TIS-75 but with an internal house number. TO-92 plastic case. N Channel, Junction type FET.
SPECIAL 5 FOR $1

FAIRCHILD BIG LED READOUTS

A big .50 inch easy to read character. Now available in either common anode or common cathode. Take your pick. Super low current drain, only 5 MA per segment typical.

FND - 510 Common Anode
FND - 503 Common Cathode

YOUR CHOICE $2 ea.

H.P. IS BACK!

No Rejects or Retests here — You Want Quality? You have it with this popular 0.3" L.E.D. by Hewlett Packard. Fits Standard I.C. Socket. Right hand DP.

5082-7740 — Common Cathode $1.25
5082-7730 — Common Anode $1.25
6 for $6.00



HEAVY DUTY TRANSFORMER

36 VAC, 3.5 AMPS. 115VAC 60 HZ. PRI. For power supplies or audio power amps. Best quality, fully enclosed. $4.95

8T97B

Signetics, TRI-State Hex Buffer. MOS and TTL interface to TRI-State Logic. Special: $1.49.

936 DTL BY ITT

Prime, house numbered. Hex Inverter 10 For $1

LM324 - QUAD 741 OP AMP - 99c

NE555 - Timers - 49c

709CH - OP AMPS - 4 For $1

GE SCR C106B1

4 AMP 200PIV. Sensitive Gate. 59c

SIGNETICS 1K P-ROM

82S129. 256 X 4. Bipolar, much faster than MOS devices. 50 NS. Tri-state outputs. TTL compatible. Field programable, and features on chip address decoding. Perfect for microprogramming applications. 16 pin DIP. With specs. $2.95 ea.

TEXAS INSTRUMENTS 8 DIGIT CALCULATOR CHIP TMSO 103NC

Four function. Same style as used in TI pocket calculators. Factory new units, with specs. 99c each.

1AMP 1000 PIV SILICON RECTIFIERS

IN4007. Factory prime devices. 10 for $1.00

741C OP AMPS

Prime, factory tested and marked. Full spec on all parameters. Not re-tested, functional only, units as sold by others.
741 CH -TO-5 8 Lead Metal Can . . . 3/$1
741CV - 8 Lead Mini Dip . . . . 4/$1

DL747 JUMBO LED READOUTS

By Litronix. .65 inch character. Common anode. Outperforms SLA-3. perfect for giant digital clocks. Only 20 MA. per segment. Special - $2.50 ea. **(Not Class II)**

SUBMINIATURE TRIMMER CAPS

Ultra stable. Range: 3.5 to 18 PF. 2 for $1

FILTER CAPS

Dual section, 2000 MFD and 1500 MFD at 30 WVDC Twist Lock. 79c ea. 3 For $2

NATIONAL VOLTAGE REGULATOR

Like 7805. 5VDC - 1 AMP output. Has TO-220 plastic power tab - 99c

IN4148 COMPUTER DIODES

Same as IN914. Factory prime. 20 for $1

FULL WAVE BRIDGE

By G. I. 1.5 AMP 800PIV - 75c

2N706A 400MHZ. TRANSISTORS

Silicon NPN. 1 WATT. TO-18 Case. 5 For $1

LARGE SIZE LED LAMPS

Similar to MV5024. Prime factory tested units. We include plastic mounting clips which are very hard to come by.
Special 5 for $1

MOTOROLA NEG. VOLTAGE REG

MC1469R. TO-66 9 Lead package. For 3 TO 30V Outputs. Provides 600 MA direct output or more by using an external power transistor. Reg. catalog $4 ea. With specs. $1.95 ea.

MOTOROLA POWER DARLINGTON $1.99

MJ3001 - NPN - 80 Volts - 10 Amps - HFE 6000 typ. To-3 Case. Ideal for power supplies, etc. We include a free 723 regulator w/schematic for power supply with purchase of the MJ3001. You get the two key parts for a DC supply for only $1.99. Regular catalog price for the MJ3001 is $3.82

MOTOROLA NEGATIVE VOLTAGE REG

MC1463R - Like our 1469R, except for negative voltage. Reg. catalog $5. Our price $1.95.

We do not sell junk. Money back guarantee on every item. WE PAY POSTAGE. Orders under $10 add 75c handling. No C.O.D. Texas Res. add 5% tax.

S. D. SALES CO.

P.O. BOX 28810
DALLAS, TEXAS 75228

MERRY CHRISTMAS AND A HAPPY NEW YEAR

LIFE, LIFE-2, which is a competitive game of life, HEXPAWN (see BYTE no. 3), and a children's literature quiz. A few of these games are worth singling out – my favorite is a buzzword generator that produces useful (?) computer jargon such as "total monitored mobility," "systematized reciprocal programming," "integrated management options" (surely a useful phrase) and "parallel digital concept." Another pair of programs, POETRY and POET, produce Haiku-like and Edgar Allen Poe-ish verse, respectively. They're somewhat deterministic, but certainly fun at first.

The preface includes a *caveat,* warning any users that the majority of the programs are written in "standard" BASIC (where "standard" apparently means DEC's BASIC), so some conversion may be necessary for other BASIC systems. I would emphasize this warning – some of DEC's BASIC features used in the games (the file handling procedures, for example) may be difficult to convert.

However, it should be clear that *101 BASIC Computer Games* is really a "Hacker's Garden of (Computer) Verse," and is certainly worth the $7.50 it costs. For ordering and bulk discount information, write Digital Equipment Corporation, Software Distribution Center, 146 Main St., Maynard MA 02154.

Man is never satisfied with what he's got – the third section of *Microprocessors* deals with limitations of microcomputers and how to push them back a bit.

Microprocessors: New Directions for Designers, *edited by Edward A. Torrero, 1975. Hayden Book Company, Rochelle Park, New Jersey. $8.95.*

Microprocessors is a paperback collection of reprints, originally published in *Electronic Design* from 1973 to 1975. Edward A. Torrero has put together a book aimed principally at the design engineer, emphasizing the practical aspects of microcomputer design.

The first section deals with an overview of the micro's impact on industry. Representative articles are "Focus on Microprocessors," a truly global assessment of the current situation and potential of microcomputers for the design engineer; and "Smart Machines in Industrial Electronics," a look at micros in numerical control and related equipment.

The second section, "Microcomputer Basics," includes articles on microprocessor selection for various applications, the capabilities and various techniques of microcomputer input and output, the choice between random logic and micros, an analysis of micro instruction sets, some basics of software construction, and an excellent worked-through example of microprocessor control of traffic lights. This section also features a detailed look at the internal operation of, and external support required for, the 8008 CPU chip.

For the hobbyist hardware expert, the book provides a fair overview of the microcomputer field; for the person more acquainted with software, it gives a good introductory look at the field from a hardware perspective.

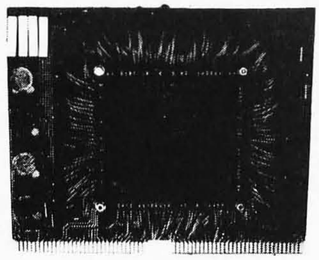
But man is never satisfied with what he's got – the third section of *Microprocessors* deals with limitations of microcomputers and how to push them back a bit. The first article, "Speeding Microcomputer Multiplication," shows how to build a CPU-complementary multiplication circuit for the IMP-16C that allows multiplication of two 16-bit unsigned operands in 23 microseconds. Using only 16 standard SSI and MSI circuits, this CPU addition reduces multiplication time by a factor of 30 from software, and by a factor of 7 from the optional National-supplied multiply instruction. Other articles include an investigation of the "microprogrammability" of some micros for special applications, construction of a system that allows a conventional mini to debug a microsystem, implementation of an external push-down stack for the 8008 to assist in interrupt handling, and specifications for providing the clock and drive signals for the 8080, something apparently missing from the 8080's application data.

The articles of the fourth and last section cover an ample range of applications. The first is a close-up of a Motorola 6800 CPU used as a controller of a simple drum printer; the second describes the use of an ACIA (Asynchronous Communications Adapter) in a minicomputer context. The last three articles deal with uses of micros in instrumentation, phase locked loop motor control, and "intelligent" networks of computers.

From this (rather exhaustive) list of articles, it should be clear that *Microprocessors* is certainly a useful book for its intended audience, the electronics design engineer. It also fills two other important functions: for the hobbyist hardware expert, it provides a fair overview of the microcomputer field; for the person more acquainted with software, it gives a good introductory look at the field from a hardware perspective, an introduction that I found very useful.

Chris Ryland
25 Follen St.
Cambridge MA 02138

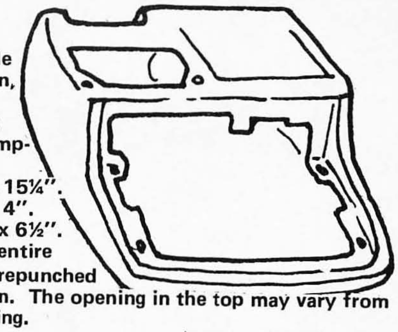
2704 x 4 BIT MEMORY BOARD



Have you ever loaded a long program into memory, only to have the power fail? Lost the entire program, huh? If so, you know the value of having a non-volatile magnetic core memory. We have acquired these core memory boards, made by Burroughs. They appear to be 52 x 52 x 4 bits wide (2704 bits x 4), for a total of 10,816 bits. Unfortunately, we don't have data, but for the modest price we are asking they are well worth the effort to figure out. 7½" x 9" x 1¾". Shipping weight 2 lbs. Less than 1/10 cent per bit!

STOCK NO. B5316 Core Memory Board $9.95 each, 4/35.

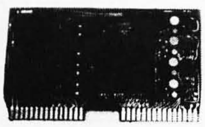
DATA INPUT KEYBOARD HOUSING



This high impact plastic housing was part of a famous name programmable desktop calculator. It is tan, has the general appearance of a typewriter case, and is ideal for many types of computer keyboards. Mounting center for the keyboard is 15¼". The readout space is 15" x 4". Overall size is 18" x 18½" x 6½". 3 lbs. Will easily house an entire TVT or micrpcomputer. Prepunched hole in rear for a muffin fan. The opening in the top may vary from the one shown in the drawing.

STOCK NO. B9176 $3.75 each, 2/7.00

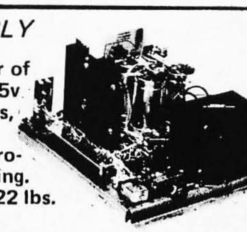
256 x 4 BIT CORE MEMORY BOARD



Another non-volatile magnetic core memory board. This one was originally made for one of America's largest calculator manufacturers. It is a complete 256 bit x 4 bits wide memory board (1024 bits). Complete with core drivers and sensors. 6¼" wide x 3¾" high, 1 lb. With limited data. QUANTITY LIMITED.

STOCK NO. B5203 $9.95 each

HEAVY DUTY POWER SUPPLY



This power supply is surplus from a maker of computer peripherals. It has outputs of +5v @ 5 amps, −5v @ 500ma, +12v @ 2.5 amps, and −12v @ 200ma. All voltages are well filtered and regulated. Ideal for your micro- or mini- computer. All are tested & working. 10½" x 13½" x 7" high, shipping weight 22 lbs.

STOCK NO. B5312 $34.95

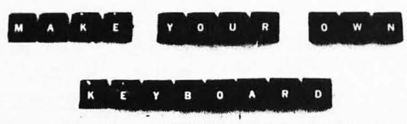
ALCO KNOBS FOR EVERY APPLICATION



Machined aluminum knobs with 2 setscrews. Clear anodized finish. Latest distributor price $ .65 to 1.05 each.

ALCO NO	O.D. x HT.	DELTA NO.	PRICE
KS 500A	.500" x .625" w/skirt	B6176	.40 ea, 10/3.50
K 700A	.750" x .620"	B6177	.40 ea, 10/3.50
K 1250A	1.250" x .750"	B6178	.50 ea, 10/4.50

KEYTOPS & SWITCHES TO MAKE YOUR OWN KEYBOARD



We have a large selection of KEYTOPS and SWITCHES, made by RAYTHEON CO. The keytops come in black, grey and white, with contrasting legends. The switches mate with the tops, and are magnetic reed switches. The following combinations are available:

54 key typewriter set, keys only,	black	K9276	2.95
54 key typewriter set, keys only,	grey	K9278	2.95
54 key TTY set, no symbols	white	K9279	2.95
54 key TTY set, with symbols	white	K9282	2.95
54 key set, keys & switches	black	K9288	30.00
54 key set, keys & switches	grey	K9290	30.00
54 TTY set, no symbols keys & Sw.	White	K9291	30.00
54 TTY set, with symbols, keys & Sw.	white	K9291	30.00
11 Key Numeric set. Keys only	Black	K9283	1.50
11 Key Numeric set, Keys only	Grey	K9284	1.50
11 Key Numeric set, Keys only	White	K9295	1.50
12 Key numeric set, Keys only	white	K9286	1.50
11 Key Num.set, keys & switches	Black	K9293	7.00
11 Key Num. set, keys & switches	Grey	K9294	7.00
11 Key Num. set, keys & switches	White	K9295	7.00
12 Key Num. set, keys & switches	white	K9296	7.50
Blank key 1½ keys wide	white	K9297A	3/.25
Blank key 2 keys wide	white	K9297B	3/.25
K9297A with switch	white	K9298A	3/2.00
K9297B with switch	white	K9298B	3/2.00

2502 AY-5-1013 UART

Brand new, tested, General Instrument AY-5-1013 UART (Universal Asynchronous Receiver/Transmitter). This 40 pin IC is pin compatable with Signetics 2536, TI's TMS6012, SMC COM2502, etc. The UART is the standard building block for parallel to serial and serial to parallel conversion, serial communication, etc. A must for TVTs, cassette interface, RS-232 interface, etc.

STOCK NO. B3119 $3.95 each

COMPUTER GRADE ELECTROLYTICS

We carry a good selection of capacitors of all types, dipped micas, ceramic disks, mica trimmers, tantalums, silvered micas, tubulars, small electrolytics, high voltage, etc. Listed below are a few of our large computer grade electrolytic capacitors.

VOLT	MFD	STOCK	PRICE	
10v	40,000	B2026	2.00 each,	6/11.00
10v	90,000	B2495	3.00 each,	3/8.00
10v	160,000	B2515	3.25 each,	4/12.00
15v	110,000	B2352	3.50 each,	3/9.00
25v	32,000	B2492	3.00 each,	4/10.00
35v	40,000	B2255	3.50 each,	3/9.00
50v	10,000	B2493	3.25 each,	4/12.00
75v	6,000	B2450	3.50 each,	4/12.00
200v	500	B2345	2.50 each,	4/9.00

1458, dual 741 type op-amp. 8 pin mini-DIP. Very versatile.
STOCK NO. B3112 $1.25 each, 5/5.00

KR-2376 KEYBOARD ENCODER ROM

This IC Keyboard Encoder is ideal for use with the above keyboard sets, or with any other keyboard. Made by General Inst., this 40 pin IC features TTL/DTL or MOS compatable outputs, external parity selection, 2 key roll-over, N-key lockout, self contained oscillator, and static charge protection of all inputs and outputs. This chip is programmed for ASCII code outputs for 88 keys in the Normal mode, using external Shift operation. In the Shift mode, it is programmed for EBCDID outputs. New, tested, with data sheets. G.I. number KR-2376-10.

STOCK NO. B3118 $9.95 each

M1568 dual 15 volt regulator IC. **Tracking regulator, 15v @ 100 ma each. 14 pin DIP. STK NO. B4481 $1.50 ea, 4/5.00**

2 MILLION BIT MEMORY DRUM

We have one 2 million bit memory drum which was removed from an experimental computer. It was manufactured by General Instrument Co. for Ultrasonic Systems Corp., Mount Laurel, N. J. It is G. I. Model MH12C-128-189, Serial No. 7816, and U.S.C. Model 620MDRA1206. It has a maximum access time of 33.4 ms, and a transfer rate in excess of 500K bits/ sec/ track. There are 108 storage tracks, expandable to 128 tracks, with a seperate head for each track. The number of bits/ track = 16,000. For more data, phone us and ask for Mel Weiss or Ray Jorgenson. 617-388-4705
$500

DELTA ELECTRONICS CO.

BOX 1, LYNN, MASSACHUSETTS 01903
Phone (617) 388-4705

MINIMUM ORDER $5.00. Include sufficient postage, excess refunded. Send for new 88 page Catalog 15, bigger than ever.

BANKAMERICARD and MASTERCHARGE now accepted, minimum charge $15.00. Please include all numbers. Phone orders accepted.

Review:

The CT-1024 Kit

James Hogenson
Box 295
Halstad MN 56548

The Southwest Tech CT-1024 is a keyboard to TV interface which displays 16 lines of 32 ASCII characters each on a TV screen. The CT-1024 may be ordered with a variety of options facilitating easy interface with any computer. The CT-1024 is a welcome compromise between one line LED displays and commercial CRT or hard copy terminals.

Data to be displayed is stored in the form of ASCII coded characters. The display circuitry scans the memory, converts the ASCII codes to 5 by 7 dot matrix characters, and then generates the video signals necessary to reproduce the characters on a TV screen.

The main board holds the character generator, synchronization and timing circuits. This double sided board, measuring 9.5 inches by 12 inches (24.1 cm by 30.5 cm) holds 42 integrated circuits and a few other parts. Three cable connectors are mounted on the main board for input, output and power supply connections. A group of connectors are provided for plug-in boards.

The memory board plugs into the main board. The memory uses 2102 type chips to store 1024 characters. The memory is divided into two pages of 512 characters each. A switch can be used to select the page, or without a switch, the pages will alternate. When one page is filled, the display automatically selects the other page.

The main board, memory board, parts for both and a 29 page manual, constitute the basic kit selling for $175. The manual includes step-by-step assembly instructions for the main board and memory board, detailed testing and calibration instructions, five and one half pages of "How it Works," connector diagrams, parts lists, and foldouts for component placement and two color foil pattern diagram. The main board circuit diagram is supplied on a separate 18 inch by 24 inch (45.7 cm by 61.0 cm) sheet. Documentation for the optional plug-in boards is provided with each of those kits.

Assembly is not difficult, but it is not a "one-evening" type of project either. It will be worth your while to take your time and be careful while putting the CT-1024 together. There is a flat charge of $25 if you have to send your main board in for repair.

Getting the CT-1024 adjusted and calibrated after assembly is as simple as turning four pots. There are no special tricks involved, and the only instrument needed is a VOM.

The CT-1024 requires 5 V at 2 A, and −5 V and −12 V at 100 mA each. If your existing system's power supply is unable to provide this, you can buy the CT-P power supply kit for $15.50; the entire supply circuit is contained on one small separate board. The PC board, components and transformer are included in the power supply kit.

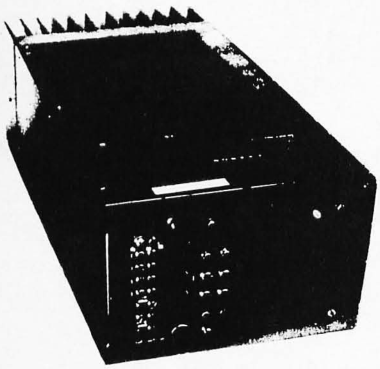
There is a wide variety of input options for the CT-1024 allowing interface with almost anything. Any TTL compatible ASCII input will be accepted. The SWTPC keyboard kit can be connected directly to the CT-1024 main board. If you do not require data ready control strobes, you can connect your microcomputer's output directly to the main board input. If this is done, however, you must program a time delay between each data transfer. This method will also be error prone if interconnect cables are more than 2 or 3 feet in length.

If the CT-1024 is used as a remote terminal, the serial interface kit is an excellent option. This kit will allow the

LOGIC POWER SUPPLIES

New Lambda transistorized, regulated, worth 2 or 3 times our price.

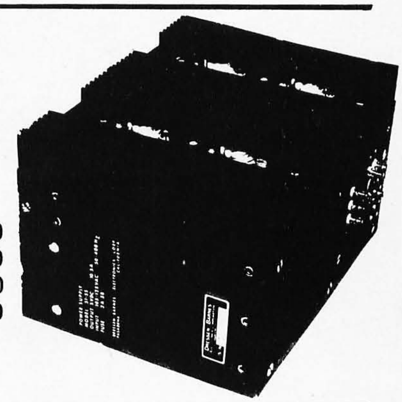
1. (LVEE – 5) OV,	5 VDC 74 amp	....	125.00
2. (LWD – 12)	12 VDC 26.5 amp	....	100.00
3. (LMB – 5)	5 VDC 3.7 amp	....	40.00
4. (LMD – 15Y)	15 VDC 9 amp	....	45.00



SOPHISTICATED LOGIC SUPPLIES

By Dressen-Barnes and NJE. Transistorized, finely filtered and regulated. 115 volt input.

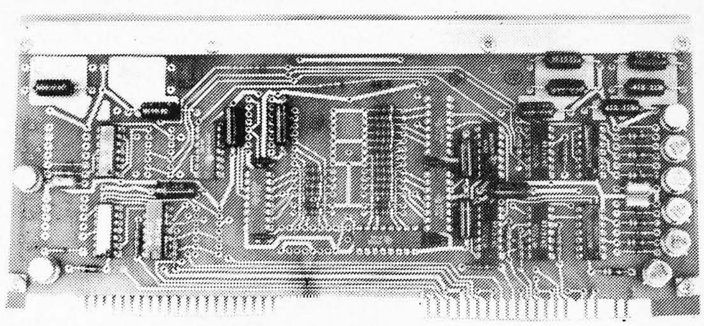
#41-5S	Output 5VDC 6 Amp	$25.00
#421-32	Output 32VDC 3.3 Amp 12 lb	20.00
#421-90	Output 90VDC 1.2 Amp 12 lb	20.00
#NJE	5VDC 34 Amp 35 lb	75.00



MEMORY SYSTEM $125.00

New memory system by Honeywell, small ... measures only 9x4x1 inches. 1024 core memory, 1024 words with 8,9,10 bits/word. Random access, with all logic, register, timing, control, core select and sense functions in one package. New, booklet of schematics and data. Looks like a good beginning for a mini-computer. Limited supply on hand.

Ship wgt 3 lbs. #SP-79 .... $125.00

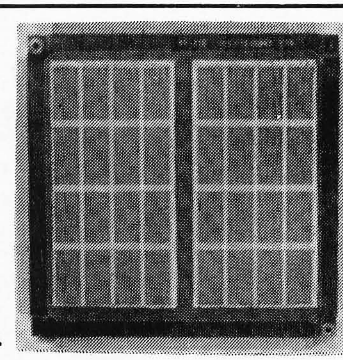


COMPUTER CAPS BRAND NEW						
1,000 μF	15 volt	$ .35		2,000 μF	35 volt	1.00
2,000	15	.50		12,000	40	2.50
1,000	25	.70		3,900	50	2.00
3,000	25	1.00		22,000	75	3.25
1,000	35	.80				

CORE MEMORY

Another brand new memory, ultra small. Measures only 4 x 4 inches with format on one plane of 32 x 32 x 16 (16,384). Only about 35 units of this on hand.

#SP-81 .... $20.00



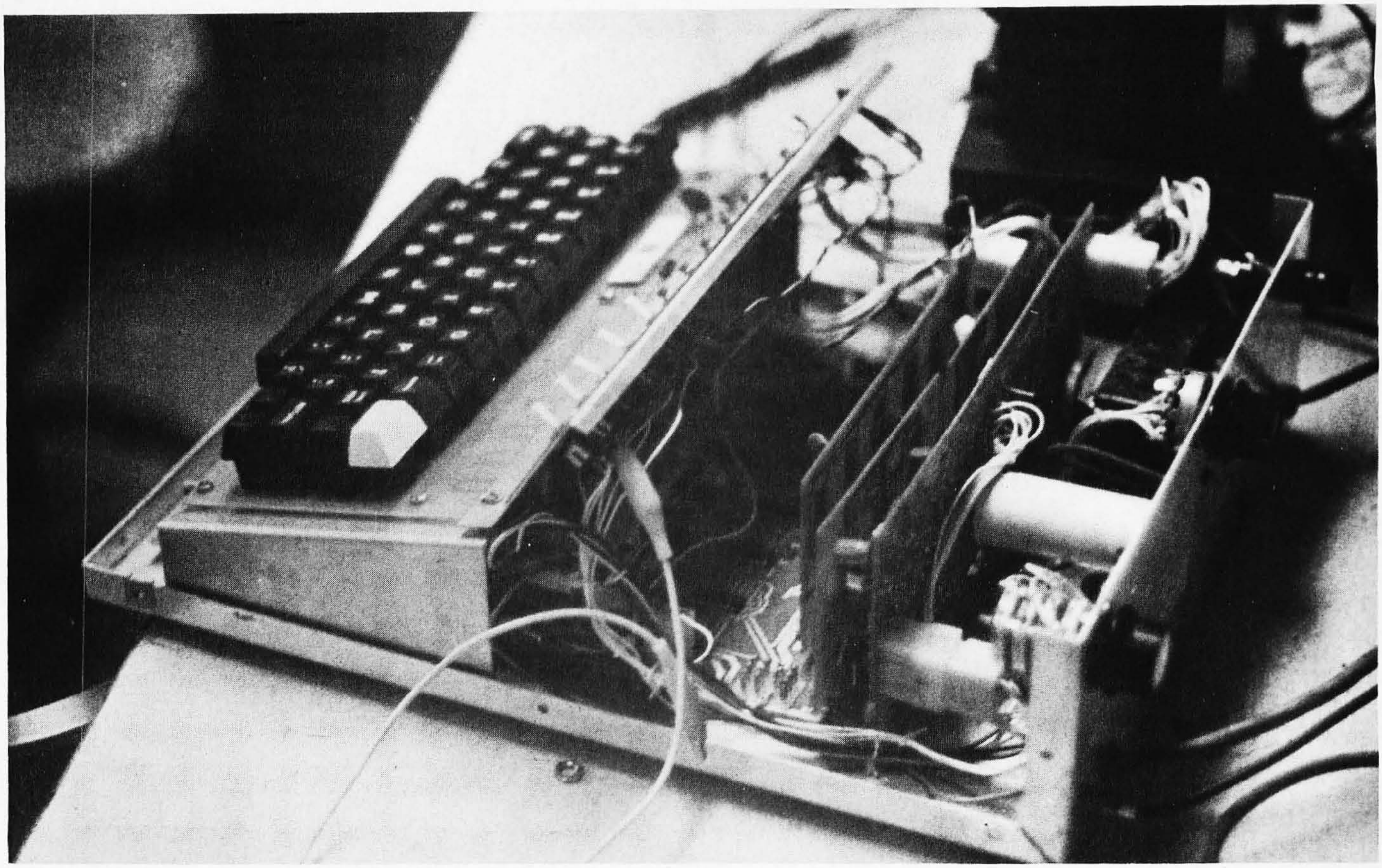
SANDERS 720 KEYBOARD $40.00

Meshna

FREE CATALOG

Please add shipping cost on above.

MESHNA PO Bx 62 E. Lynn Mass. 01904



Southwest Tech CT-1024 video terminal . . . built on a homemade base. Row of switches along the top is for extra functions.

terminal to communicate via a three wire system, phone line, or magnetic tape storage system.

The serial interface will convert parallel data to serial data and will produce not only the serial data, but start, stop and parity bits as well. The serial interface will both transmit and receive, thus making the CT-1024 a complete terminal. Standard speed is 110 baud. Adding optional parts to the PC board will provide 150, 300, 600 or 1200 baud.

When the serial interface is used, the keyboard is connected to the serial interface board. In the echo mode, data is transmitted to the receiver, but is not displayed on the screen until it is transmitted back to the terminal. Otherwise data is transmitted and simultaneously displayed.

The input and output connections are RS-232 compatible and will attach directly to most couplers and data sets. To record on or play back from magnetic tape, it will be necessary to build an FSK circuit which is not included on the serial interface board.

The serial interface board measuring 3.375 inches by 9.5 inches (8.6 cm by 24.1 cm) plugs into the main board. This kit including thorough documentation sells for $39.95.

A parallel interface board is available for faster data transfer between the CT-1024 and your microcomputer. The parallel interface board provides the handshaking, buffering and timing necessary to transfer data properly at high speed in a number of system configurations. The various operating modes are programmed by jumper wires. The versatility of the parallel interface board makes the CT-1024 compatible with any computer system – from home brew hobby systems to large scale commercial systems.

High noise immunity in the parallel interface is provided by a special circuit on the strobe lines and on all inputs from the data bus. For maximum flexibility, all data and strobe lines from the IO bus can be selectively inverted by programming jumpers on the PC board. The keyboard can be directed to just print data on the screen, print data on the screen and load the data onto the output bus, or just load data onto the output bus. The latter option will allow you to have all data echoed back for verification.

The interface input and output bus lines can be used separately or parallel for bidirectional systems. To make interfacing even easier, the data flow control lines can be either strobed or operated in a demand/response mode. All of these options are selected by inserting jumper wires on the PC board.

The parallel interface board measures 4 inches by 9.5 inches (10.2 cm by 24.1 cm) and plugs into the main board. Both parallel and serial interface boards cannot be used at the same time. The parallel interface kit is priced at $22.95.

The CT-1024 produces a video signal output which can be viewed on a video monitor or a modified television set. Don't let the "modified" scare you away because the modification is quite simple.

I picked up an old tube type junker for $15. I inserted a 10 pF capacitor as mentioned in the instructions, left out the rest, but placed a 6.8 kO resistor and a 180 pF disc capacitor in parallel between the CT-1024 video output and the TV video input. I disconnected the IF stage from the grid of the video output tube and connected the CT-1024 at that point. The only way to find out what modifications are necessary on your set is to start experimenting. The television modifications necessary will vary from one set to the next.

The CT-1024 includes another convenience: a cursor. The cursor is a little blinking square displayed on the screen which shows where the next character entered will be displayed. An optional switch will stop the blinking action or blank out the cursor if desired.

The cursor generation circuitry is included on the main board. An optional cursor control kit will provide a wider range of controls over the cursor. The manual cursor control kit will provide switch debouncing for your choice of switches. The use of these switches or keys will allow you to move the cursor up, down, left or right. "Home up" will return the cursor to the upper left corner of the screen. "Erase to end of line" will erase anything between the cursor and the end of the line. "Erase to end of frame" will erase anything between the cursor and the lower right corner of the frame.

The cursor control board plugs into the main board. The cursor control switches connect via wires to the cursor control board. The manual cursor control kit is available for $11.50.

If you are not presently in a position to fork out the bills for a good Teletype, join the club. And get a CT-1024. Or for those of you who have a Teletype rig, how about all those times you read a printout once, then threw it away and tried again? A TV terminal might be worth the consideration. There are lots of other uses for a TV typewriter without the computer, too. Whatever your application might be, it seems that there are a lot of satisfied CT-1024 users. ■

BYTE reader service

To get further information on the products advertised in this issue of BYTE merely tear, rip, or snip out this advertiser index, fill out the data at the bottom of the page, circle the appropriate numbers, and send the works to BYTE, 70 Main St, Peterborough NH 03458. Readers get extra Brownie Points for sending for information since this encourages advertisers to keep using BYTE – which in turn brings you a bigger BYTE.

ADVERTISER INDEX

***Reader service inquiries not solicited. Correspond directly with company.**

Reader's Service
BYTE
Green Publishing Inc
70 Main St
Peterborough NH 03458

Please print or type.

JANUARY 1976

BYTE acquired via
☐ Subscription
☐ Newsstand
☐ Stolen

Name ______________________________

Address ______________________________

City ____________ State ________ Zip ________

Coupon expires in 60 days . . .

THE BYTE QUESTIONNAIRE

BYTE is dedicated to the needs of its readership. In order to better gauge matters of editorial policy and content, as well as to give our advertisers some "hard facts," we publish this questionnaire.

What applications do you have in mind for your own personal computer system?

Do you presently have a computer?

If you own or use a personal computer system, what is its CPU chip or main frame computer design?

Have you had any "on the job" training using computers as a tool for some purpose?

Do you know what a computer language is? Would you list the computer languages (if any) which you know?

These questions are a short form "letter to the editor." If you have additional comments, don't hesitate to write! Send completed questionnaires to BYTE, Dept. Q, Peterborough NH 03458.

Feel free to photocopy this page if you wish to keep your BYTE intact.

THINKING ABOUT A "6800" TYPE COMPUTER?

It seems that a great many people came to the same conclusion that we did here at SwTPC. The M6800 is an outstanding processor and makes a great computer — "BUT" — Not all computers using the M6800 processor are the same. May we suggest that you consider the following features when you make your choice.

IT IS A COMPLETE 6800 SYSTEM?

You cannot get all of the advantages of the 6800 system with only the processor chip. Unless the whole 6800 family of chips is used you cannot possibly get all of the versitility and superior performance that the system is capable of providing. If for instance the design does not use the MC6820 parallel and the MC6850 serial integrated circuits for interfacing, you lose the programmable interface feature that makes it so easy to interconnect the computer system with outside devices such as terminals, printers, disks, etc.

IS THE SOFTWARE COMPATABLE OR UNIQUE?

If the design does not use the "Motorola" Mikbug® ROM, then the software and programs that will run on the system are probably unique to that particular brand of computer. SwTPC uses the standard Motorola MCM6830L7 ROM. This provides automatic loading and an operating system that is compatable with other systems using the standard widely sold Motorola evaluation set. As an owner of our 6800 computer system, you are eligible for membership in the Motorola Users Group. If you join you have access to a library of programs that will run on your system. Editor and assembler programs are available directly from SwTPC.

CAN THE SYSTEM BE EXPANDED AT A REASONABLE COST?

Some of the limited systems being offered at lower prices can be expanded only with difficulty. Check the amount of memory that can be added and at what cost. How many additional interfaces can be added, if any. How much of the above can be run off of the power supply provided with the system? The SwTPC 6800 can be expanded up to 16K words of memory in the standard cabinet and with the power supply provided. It may also be expanded up to eight interface (I/O) boards for external devices by simply plugging in the cards. Memory is $125.00 for each 4,096 words of expansion and interface cards are only $35.00 for serial or parallel types.

Memory expansion will be essential if you ever intend to use a resident assembler, or higher level languages such as APL or BASIC on your system. Assembler programs typically require a minimum of 4,096 words of memory and higher level languages require even more.

HOW DO YOU ENTER AND READ DATA?

Let's hope it is by way of a TTY, or video terminal. No one with a serious computer application would consider attempting to enter data from a switch and status light console. These may be educational, but they sure aren't practical. Calculator keypads and digital readouts are not much better. There is no substitute for a full alphanumeric keyboard and terminal system display for serious work.

Mikbug® is a registered trademark of Motorola Inc.

SwTPC 6800

Computer System
with serial interface and 2,048 Words of memory $450.00



☐ Enclosed is $450 for my SwTPC Computer Kit ☐ Send Data

☐ or BAC ____________ # ____________

☐ or MC ____________ Ex Date ____________

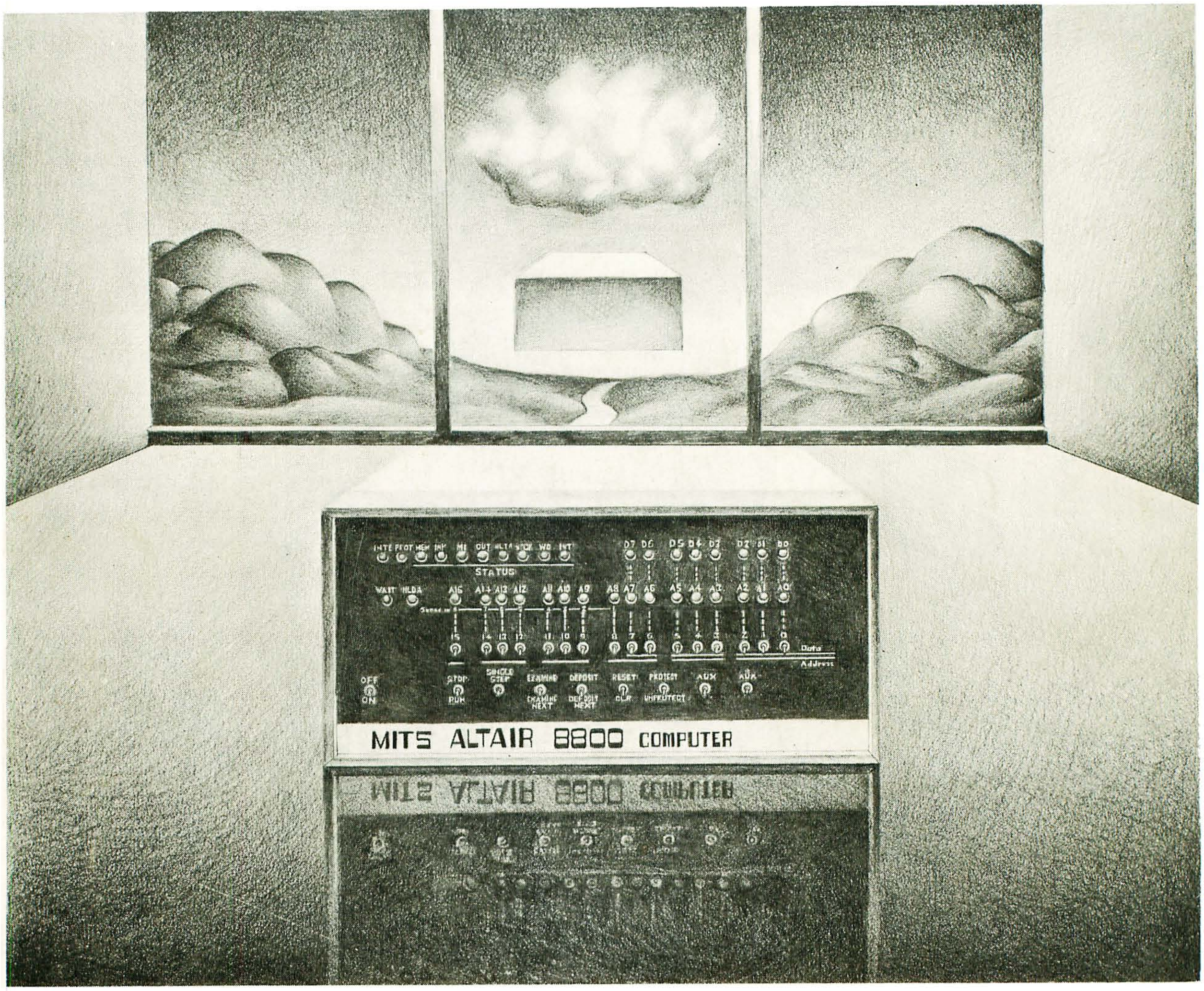
NAME ____________

ADDRESS ____________

CITY ____________ STATE ____________ ZIP ____________

Southwest Technical Products Corp., Box 32040, San Antonio, Texas 78284

Created by Man.



The Affordable Computer.